

**ANALISIS KINERJA PROTOKOL 6LOWPAN
PADA JARINGAN SENSOR NIRKABEL
DENGAN TOPOLOGI JARINGAN GRID dan
TOPOLOGI JARINGAN RANDOM MENGGUNAKAN COOJA
SIMULATOR**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Andre Hadi Saputra
NIM: 145150200111063



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

ANALISIS KINERJA PROTOKOL 6LOWPAN PADA JARINGAN SENSOR NIRKABEL
DENGAN TOPOLOGI JARINGAN GRID dan TOPOLOGI JARINGAN RANDOM
MENGUNAKAN COOJA SIMULATOR

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer


Disusun Oleh :
Andre Hadi Saputra
NIM: 145150200111063

Skripsi ini telah diuji dan dinyatakan lulus pada
28 Desember 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II


Ir. Primantara Hari Trisnawan, M. Sc.
NIP: 19680912 199403 1 002


Fariz Andri Bakhtiar, S.T., M.Kom.
NIK: 201709840314 1 001

Mengetahui
Ketua Jurusan Teknik Informatika




Tri Astolokurniawan, S.T., M.T., Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 09 Januari 2019



Andre Hadi Sabutra

NIM: 145150200111063

KATA PENGANTAR

Puji syukur atas karunia Allah SWT yang Maha Pengasih lagi Maha Penyayang karena dengan rahmat dan karuniaNya, penulis dapat menyelesaikan skripsi yang berjudul ***“ANALISIS KINERJA PROTOKOL 6LOWPAN PADA JARINGAN SENSOR NIRKABEL DENGAN TOPOLOGI JARINGAN GRID dan TOPOLOGI JARINGAN RANDOM MENGGUNAKAN COOJA SIMULATOR”*** sebagai salah satu syarat memperoleh gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawijaya.

Penulis menyadari bahwa dalam menyelesaikan skripsi ini tidak lepas dari peran berbagai pihak yang telah memberi semangat, doa, bimbingan serta kritik dan nasehatnya. Dalam kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada :

1. Allah SWT yang telah memberikan rahmat dan karuniaNya dalam pelaksanaan skripsi ini.
2. Bapak Ir. Primantara Hari Trisnawan, M. Sc selaku dosen pembimbing I yang telah sabar dalam membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Fariz Andri Bakhtiar, S.T., M.Kom. selaku dosen pembimbing II yang telah sabar dalam membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Agus Wahyu Widodo, S.T, M.Sc selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
6. Ayah Muhadi dan ibu Munaiyah atas segala dukungan, nasehat, motivasi dan doa yang telah dipanjatkan untuk menyemangati penulis dalam menyelesaikan skripsi ini.
7. Seluruh teman – teman Fakultas Ilmu Komputer angkatan 2014 Universitas Brawijaya yang telah memebrikan saran, memotivasi penulis untuk menyelesaikan skripsi ini.
8. Semua pihak yang belum tersebutkan diatas, atas segala *support*, motivasi dan doa dalam menyelesaikan skripsi ini.

Malang, 09 Januari 2019

Penulis

andre.saputra1790@gmail.com

ABSTRAK

Andre Hadi Saputra, Analisis Kinerja Protokol 6LoWPAN Pada Jaringan Sensor Nirkabel Dengan Topologi Jaringan Grid Dan Topologi Jaringan Random Menggunakan Cooja Simulator

Pembimbing: Ir. Primantara Hari Trisnawan, M. Sc dan Fariz Andri Bakhtiar, S.T., M.Kom.

IPv6 over Low power Wireless Personal Area Network (6LoWPAN), adalah protokol berbasis IPv6 over low power yang memungkinkan perangkat dengan konsumsi daya kecil seperti sensor tetap dapat terkoneksi dan ikut berpartisipasi dalam *Internet of Things (IoT)* dengan menggunakan jaringan sensor nirkabel maka dibutuhkan skalabilitas dan interoperabilitas yang luas. Pada penelitian ini dilakukan simulasi penggunaan protokol 6LoWPAN pada jaringan sensor nirkabel atau *Wireless Sensor Network (WSN)* menggunakan Cooja simulator lalu di analisis menggunakan aplikasi *Wireshark* dan hasil dari *Mote Output* yang dihitung secara manual dengan meninjau parameter *Quality of Service (QoS)* yaitu *throughput*, *delay* dan *jitter* untuk mengetahui kinerja 6LoWPAN. Cooja Simulator digunakan karena membangun jaringan propagasi yang asli memakan waktu dan biaya yang besar. Dari hasil simulasi protokol 6LoWPAN pada WSN menggunakan topologi grid dan topologi random yang memiliki jumlah node yang berbeda-beda dan *Routing Protocol for Low Power and Lossy Network (RPL)*. Perolehan *throughput*, *delay* dan *jitter* dihitung secara manual kecuali *throughput* yang menggunakan *wireshark* untuk mendapatkan nilai *throughput*. *Throughput* tertinggi diperoleh topologi jaringan grid dengan jumlah node 10 yaitu 435 bps, *delay* terendah diperoleh topologi jaringan grid dengan jumlah node 10 yaitu 174 ms, *jitter* terendah diperoleh topologi jaringan grid dengan jumlah node 10 yaitu 1,795 ms. Dapat disimpulkan kinerja protokol 6LoWPAN pada jaringan sensor nirkabel dengan topologi grid lebih baik daripada topologi random, karena pada topologi grid mendapat nilai *throughput* yang tinggi dengan nilai *delay* dan *jitter* yang rendah.

Kata kunci : *Internet of Things, 6LoWPAN, WSN, QoS, wireshark, Routing Protocol for Low Power and Lossy Network*

ABSTRACT

Andre Hadi Saputra, 6LoWPAN Protocol Performance Analysis on Wireless Sensor Networks with Grid Network Topology and Random Network Topology Using Cooja Simulator

Pembimbing: Ir. Primantara Hari Trisnawan, M. Sc dan Fariz Andri Bakhtiar, S.T., M.Kom.

IPv6 over Low Power Wireless Personal Area Network (6LoWPAN), is an IPv6-based over low power protocol that allows devices with small power consumption such as sensors to remain connected and participate in Internet of Things (IoT) using wireless sensor networks requires extensive scalability and interoperability. In this study a simulation of the use of 6LoWPAN protocol on wireless sensor networks (Wireless Sensor Network (WSN) using Cooja simulators was analyzed using the Wireshark application and the results of Mote Output were calculated manually by reviewing the Quality of Service (QoS) parameters, such as throughput, delay and jitter to find out the performance of 6LoWPAN. Cooja Simulator is used because building a genuine propagation network is time consuming and costly. From the simulation results using grid topology and random topology that have different number of nodes and Routing Protocol for Low Power and Lossy Network (RPL). The acquisition of throughput, delay and jitter is calculated manually except throughput using Wireshark to get throughput. The highest throughput obtained by grid network topology with the number of nodes 10 is 435 bps, the lowest delay is obtained by grid network topology with the number of nodes 10 is 174 ms, the lowest jitter is grid network topology with the number of nodes 10 which is 1.795 ms. It can be concluded that the performance of the 6LoWPAN protocol on wireless sensor networks with grid topology is better than random topologies, because the grid topology gets high throughput with low delay and jitter values.

Keywords : *Internet of Things, 6LoWPAN, WSN, QoS, wireshark, Routing Protocol for Low Power and Lossy Network*

DAFTAR ISI

PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	Error! Bookmark not defined.
KATA PENGANTAR.....	iii
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 <i>Wireless Sensor Network</i> (WSN)	6
2.2 IEEE 802.15.4 dan 6LoWPAN	6
2.3 Cooja Simulator.....	8
2.4 Wireshark.....	10
2.5 Routing Protocol for Low Power and Lossy Network (RPL).....	11
2.6 Quality of Service (QoS)	14
BAB 3 METODOLOGI	15
3.1 Studi Literatur	16
3.2 Perancangan	16
3.3 Simulasi dan Pengujian	17
3.4 Analisis	17

3.5 Pengambilan Kesimpulan.....	17
BAB 4 PERANCANGAN DAN IMPLEMENTASI	18
4.1 Perancangan	18
4.1.1 Analisis Kebutuhan.....	18
4.1.2 Perancangan Topologi.....	19
4.1.3 Perancangan Skenario Pengujian.....	20
4.2 Implementasi	21
4.2.1 Membuat Topologi dan Menjalankan Simulasi	21
4.2.2 Mengelola Hasil Simulasi Cooja Simulator.....	27
BAB 5 HASIL DAN PEMBAHASAN	29
5.1 Hasil Pengujian.....	29
5.1.1 Hasil Throughput.....	29
5.1.2 Hasil Delay.....	30
5.1.3 Hasil Jitter.....	30
5.2 Analisis	31
5.2.1 Analisis Perbandingan <i>Throughput</i>	31
5.2.2 Analisis Perbandingan <i>Delay</i>	33
5.2.3 Analisis Perbandingan <i>Jitter</i>	34
BAB 6 KESIMPULAN.....	37
6.1 Kesimpulan.....	37
6.2 Saran	38
DAFTAR PUSTAKA.....	39

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	5
Tabel 4.1 Perangkat Keras yang Digunakan	18
Tabel 4.2 Perangkat Lunak yang Digunakan	19
Tabel 4.3 Parameter Simulasi.....	20
Tabel 4.4 Konfigurasi Topologi Skenario 10 Node	21
Tabel 4.5 Konfigurasi Topologi Skenario 20 Node	22
Tabel 4.6 Konfigurasi Topologi Skenario 40 Node	23
Tabel 4.7 Konfigurasi Topologi Skenario 80 Node	25
Tabel 5.1.1 Pengujian Parameter <i>Throughput</i>	29
Tabel 5.2 Pengujian Parameter <i>Delay</i>	30
Tabel 5.3 Pengujian Parameter <i>Jitter</i>	30
Tabel 5.4 Analisis Perbandingan <i>Throughput</i>	31
Tabel 5.5 Analisis Perbandingan <i>Delay</i>	33
Tabel 5.6 Analisis Perbandingan <i>Jitter</i>	35



DAFTAR GAMBAR

Gambar 2.1 Layer Model 6LoWPAN	7
Gambar 2.2 Antar Muka Cooja Simulator	9
Gambar 2.3 Radio Messages Tools	10
Gambar 2.4 Tampilan Antar Muka Wireshark	10
Gambar 2.5 Bagan Komunikasi Motes Baru Dan Motes Lama Pada DODAG	12
Gambar 2.6 Illustrasi Terbentuknya DODAG	13
Gambar 3.1 Metodologi Penelitian	15
Gambar 3.2 Perancangan Sistem	16
Gambar 4.1 Topologi Grid dan Topologi Random	20
Gambar 4.2 Topologi Grid dan Topologi Random Menggunakan 10 node	22
Gambar 4.3 Topologi Grid dan Topologi Random Menggunakan 20 node	23
Gambar 4.4 Topologi Grid dan Topologi Random Menggunakan 40 node	24
Gambar 4.5 Topologi Grid dan Topologi Random Menggunakan 80 node	27
Gambar 4.6 Capture File Properties Topologi Grid Menggunakan 10 Node	27
Gambar 4.7 Capture File Mote Output	27
Gambar 4.8 Rata-Rata Delay	28
Gambar 4.9 Total Jitter	28
Gambar 5.1 Analisis Perbandingan <i>Throughput</i>	32
Gambar 5.2 Analisis Perbandingan <i>Delay</i>	34
Gambar 5.3 Analisis Perbandingan <i>Jitter</i>	35

BAB 1 PENDAHULUAN

1.1 Latar belakang

Seiring berkembangnya infrastruktur internet, *Internet of Things* (IoT) menjadi babak baru bagi kehidupan manusia. Pada dasarnya IoT mengacu pada benda yang bisa diidentifikasi secara unik sebagai representasi virtual dalam struktur berbasis internet. Selain *smartphone* dan komputer, semua perangkat lain kini bisa tersambung pada jaringan global dan lokal yang bertujuan untuk memperluas manfaat dari konektivitas internet secara terus-menerus seperti kemampuan berbagi data, *remote-control*, dan lainnya (Jiang, Zhang, dan Wang, 2013).

Dalam memaksimalkan dan menopang suatu infrastruktur IoT, maka diperlukan mekanisme dalam pengumpulan data yang saling berkomunikasi secara nirkabel dengan menggunakan sensor. *Wireless Sensor Network* (WSN) merupakan suatu jaringan nirkabel yang mempunyai kumpulan *node* sensor yang tersebar di suatu area tertentu. Setiap *node* sensor mempunyai kemampuan berkomunikasi dengan *node* sensor lainnya dan untuk mengelolah data (Arrosyid, Tjahjono, dan Sunarno, 2009).

Dengan kemajuan teknologi WSN yang sekarang ini bahwa proses pengolahan data tidak hanya berada di dalam satu Laptop/ *Personal Computer* (PC) di satu bidang tertentu, namun juga ada di dalam beberapa aplikasi teknologi di berbagai bidang lainnya seperti di dalam aplikasi kesehatan, militer, mesin-mesin industri, dan pelayanan sipil yang telah terintegrasi dengan sensor. Sehingga skalabilitas menjadi masalah besar (Anggraini, Irawati, dan Mayasari, 2014).

Pada awal tahun 1990-an organisasi *Internet Engineering Task Force* (IETF) mulai sadar bahwa nantinya *routing protocol* Internet Protocol versi 4 (IPv4) akan mengalami keterbatasan dalam penyediaan sebuah alamat *Internet Protocol* (IP) dan mulai mencari *Internet protocol* pengganti yang bisa menyediakan jumlah alamat IP lebih banyak. Sehingga pada tahun 1994 IETF mempunyai gambaran yang jelas mengenai peralihan menuju Internet Protocol versi 6 (IPv6) (Amer, 2012). Protokol *IPv6 over Low Power Wireless Personal Area Networks* (6LoWPAN) ini dapat mengkoneksikan perangkat-perangkat dengan konsumsi daya kecil dan ikut berpartisipasi dalam IoT. *Internet protokol* ini menerapkan *header compression mechanism* dan enkapsulasi yang memungkinkan paket dapat diterima dan dikirim melalui jaringan berbasis IEEE 802.15.4 (Schonwalder, 2010).

Routing Protocol for Low Power and Lossy Network (RPL) mempunyai solusi perutean proaktif yang dirancang khusus untuk WSN. Agar protokol routing berhasil maka setiap *node* dalam jaringan harus membantu mempertahankan rute terbaik ke tujuan yang ditentukan. Ini dilakukan dengan secara berkala memperbarui *node* tetangga dengan informasi yang mereka

butuhkan untuk menentukan rute terbaik, melalui paket yang dikirim secara nirkabel ke *node* tetangga mereka (Tsvetkov, 2011).

Protokol 6LoWPAN pada topologi jaringan grid dan topologi jaringan random ini dapat memungkinkan konsumsi perangkat daya kecil dapat terkoneksi dalam area yang jangkauannya kecil, oleh karena itu topologi jaringan grid dan topologi jaringan random dipilih untuk dibandingkan mana yang terbaik dalam menghadapi area yang mempunyai *traffic* yang rendah dalam jangkauan yang kecil (Satrio, Munadi, dan Mayasari, 2016).

Cooja Simulator merupakan tool yang telah digunakan secara luas dalam domain WSN yang juga telah dimasukkan didalam Avrora software dan MSPSim untuk melakukan *cycle-exact math emulation* (Roussel, 2016). Selain itu penggunaan simulator jaringan ini diperlukan karena dapat mereproduksi lingkungan propagasi nirkabel merupakan hal yang nyaris mustahil. Begitu juga dengan perangkat nirkabel atau penggunaan sensor nyata dinilai kurang praktis dan sangat mahal (Grag, 2015).

Maka dengan hadirnya 6LoWPAN sebagai *Internet protocol* untuk jaringan sensor nirkabel yang menawarkan interoperabilitas dan skalabilitas yang luas, diperlukan analisis mengenai kinerja internet protokol tersebut. Pada penelitian ini dilakukan analisis kinerja dengan mensimulasikan protokol 6LoWPAN pada jaringan sensor nirkabel dengan menggunakan routing protokol RPL pada topologi jaringan grid dan topologi jaringan random menggunakan Cooja Simulator. Topologi jaringan grid memperhitungkan antara letak *node* sensor, sedangkan topologi jaringan random tanpa memperhitungkan letak antara *node* sensor. Kinerja 6LoWPAN akan diuji menggunakan indeks *Quality of Services* (QoS) dengan tiga parameter yaitu *throughput*, *delay*, dan *jitter*. *Throughput* dapat menghasilkan informasi jumlah data yang dikirimkan dari node asal ke node tujuan dalam satuan waktu; *Delay* dapat menghasilkan informasi perbedaan rentang waktu kedatangan semua paket ke node tujuan; *Jitter* menghasilkan informasi banyak variasi delay selama tranmisi yang dilakukan untuk mengirimkan paket. Parameter QoS tersebut mengacu pada kemampuan jaringan untuk memberikan layanan yang baik pada lalu lintas jaringan dari berbagai teknologi (Caudhary, 2012).

1.2 Rumusan masalah

Adapun perumusan masalah yang diambil dari latar belakang yang diuraikan sebelumnya adalah sebagai berikut:

1. Bagaimana membangun lingkungan simulasi WSN berbasis 6LoWPAN menggunakan Cooja Simulator?
2. Bagaimana membangun topologi jaringan sensor nirkabel berbasis grid dan random dengan *Routing Protocol Low Power and Lossy Network* (RPL)?

3. Bagaimana mengukur parameter Quality of Service (QoS) *jitter*, *delay*, dan *throughput* untuk setiap topologi grid dan random?
4. Bagaimana kinerja 6LoWPAN pada topologi berbasis grid dan random yang disimulasikan dengan cooja simulator?

1.3 Tujuan

Adapun tujuan yang dicapai pada penelitian ini adalah membangun simulasi WSN berbasis 6LoWPAN pada cooja simulator, serta mengukur kinerja protokol 6LoWPAN pada topologi jaringan grid dan topologi jaringan random menggunakan parameter *throughput*, *delay*, dan *jitter*.

1.4 Manfaat

Adapun manfaat yang dicapai pada penelitian ini adalah dapat mengetahui cara membangun simulasi protokol 6LoWPAN pada jaringan sensor nirkabel menggunakan cooja simulator dan dapat mengetahui kinerja protokol 6LoWPAN pada WSN.

1.5 Batasan masalah

Supaya menghindari perluasan dari permasalahan tugas akhir ini, maka analisis penelitian ini akan dibatasi yang mencakup hal-hal berikut:

1. Node yang dipakai pada simulasi jaringan sensor nirkabel merupakan node yang tidak berpindah-pindah
2. Simulasi jaringan sensor nirkabel dilakukan menggunakan Cooja Simulator
3. Analisis kinerja dilakukan menggunakan aplikasi Wireshark dan Mote output yang digunakan untuk menghitung *delay* dan *jitter* secara manual

1.6 Sistematika pembahasan

Pada bagian sistematika pembahasan akan dijelaskan uraian yang meliputi beberapa bab sebagai berikut :

BAB I Pendahuluan

Bab ini berisi tentang latar belakang yang menjadi alasan pemilihan judul, perumusan masalah, tujuan penelitian, manfaat penelitian, pembatasan masalah, dan sistematika penelitian.

BAB II Landasan Kepustakaan

Bab ini menguraikan tentang kajian pustaka terkait penelitian terdahulu yang berhubungan dengan topik protokol 6LoWPAN, serta menjelaskan landasan teori yang berhubungan dengan permasalahan yang dirumuskan sebagai acuan dalam membahas permasalahan yang terjadi pada analisis kinerja protokol 6LoWPAN pada topologi grid dan random.

BAB III Metodologi

Bab ini menguraikan tentang metode dan langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri dari studi literatur, perancangan, simulasi, pengujian, analisis serta pengambilan keputusan dan saran.

BAB IV Perancangan dan Implementasi

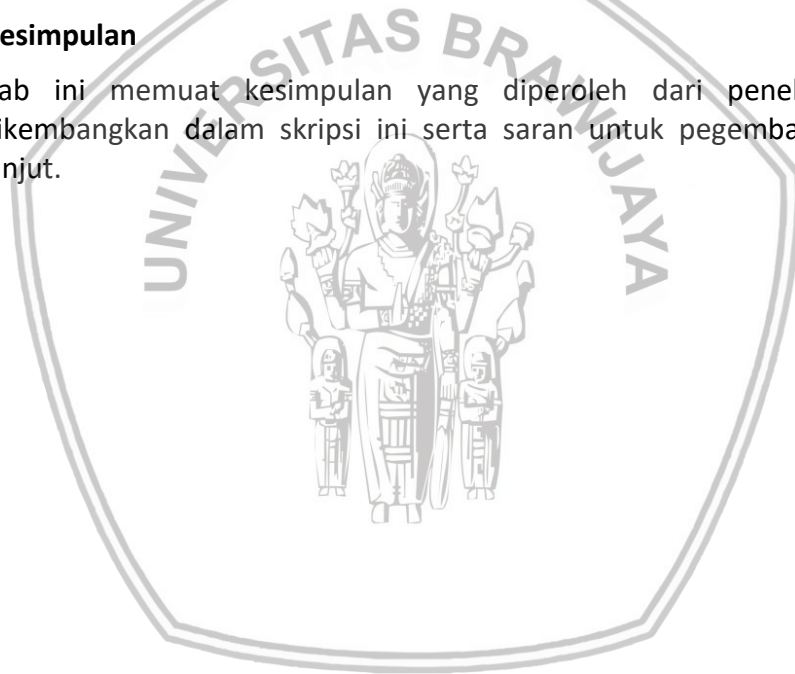
Bab ini berisi tentang perancangan dan implementasi lingkungan pengujian protokol 6LoWPAN yang menggunakan topologi grid dan random.

BAB V Hasil dan Pembahasan

Bab ini berisi mengenai pembahasan tentang analisis terhadap protokol 6LoWPAN pada topologi grid dan random. Hasil pengujian dipaparkan secara lengkap dalam bentuk perbandingan dan analisis dilakukan berdasarkan hasil pengujian yang telah dilakukan.

BAB VI Kesimpulan

Bab ini memuat kesimpulan yang diperoleh dari penelitian yang dikembangkan dalam skripsi ini serta saran untuk pengembangan lebih lanjut.



BAB III Metodologi

Bab ini menguraikan tentang metode dan langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri dari studi literatur, perancangan, simulasi, pengujian, analisis serta pengambilan keputusan dan saran.

BAB IV Perancangan dan Implementasi

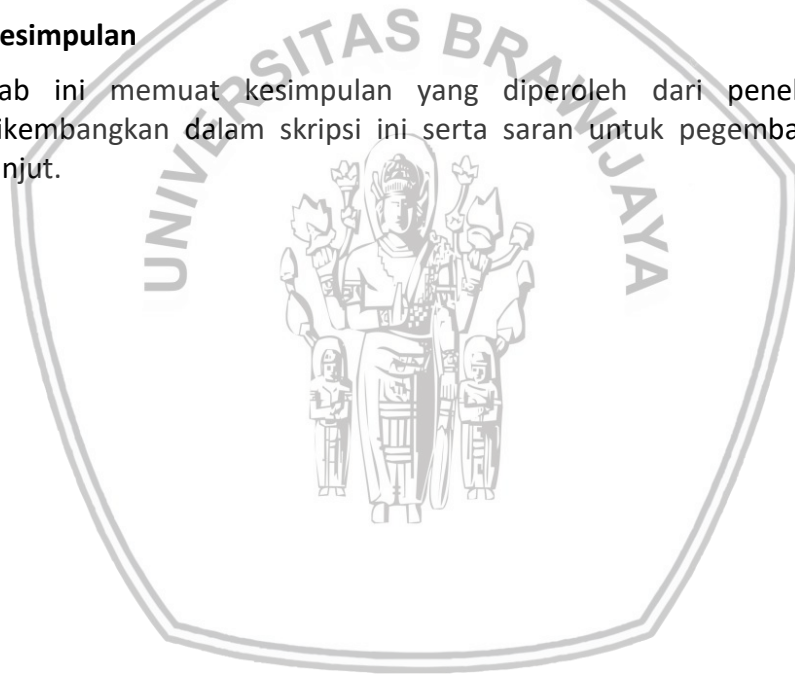
Bab ini berisi tentang perancangan dan implementasi lingkungan pengujian protokol 6LoWPAN yang menggunakan topologi grid dan random.

BAB V Hasil dan Pembahasan

Bab ini berisi mengenai pembahasan tentang analisis terhadap protokol 6LoWPAN pada topologi grid dan random. Hasil pengujian dipaparkan secara lengkap dalam bentuk perbandingan dan analisis dilakukan berdasarkan hasil pengujian yang telah dilakukan.

BAB VI Kesimpulan

Bab ini memuat kesimpulan yang diperoleh dari penelitian yang dikembangkan dalam skripsi ini serta saran untuk pengembangan lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bagian bab ini akan membahas teori pada penelitian yang terdiri dari dasar teori dan tinjauan pustaka dari penelitian terdahulu. Pada penelitian ini kajian pustaka diambil dari penelitian yang dapat dilihat pada Tabel 2.1 berikut.

Tabel 2.1 Kajian Pustaka

No	Nama penulis, Tahun, Judul	Persamaan	Perbedaan	
			Penelitian terdahulu	Rencana penelitian
1	Harka, [2010], Perancangan dan implementasi <i>Wireless Sensor Network</i> (WSN) Berbasis IEEE 802.15.4/ZIGBEE	Menggunakan WSN pada <i>low energy consumption IP protocol</i>	Menggunakan Zigbee dan <i>node sensor hardware</i>	Menggunakan 6LoWPAN dan cooja simulator
2	Dwima Indraini, Ratna, [2014], Analisis Dan Simulasi <i>Wireless Sensor Network</i> (WSN) Untuk Komunikasi Data Menggunakan Protokol Zigbee	Menganalisis komunikasi data pada WSN	Menggunakan ZIGBEE dan OPNET MODELER dan berfokus pada analisis kinerja	Berfokus pada analisis kinerja protokol 6LoWPAN menggunakan cooja simulator
3	Kevin Roussel, Ye-Qiong Song, Olivier Zendra, [2016], Using Cooja for WSN Simulations Some New User and Limits	Simulasi WSN menggunakan Cooja	Menganalisis sejauh mana Cooja dapat digunakan untuk simulasi WSN	Berfokus pada kinerja 6LoWPAN pada WSN
4	Wawan Darmawan, [2018], Analisis Performa Protokol 6LoWPAN pada Jaringan Sensor Nirkabel Menggunakan Cooja Simulator	Analisis Protokol 6LoWPAN pada WSN Menggunakan Cooja Simulator	Menggunakan Topologi cluster	Menggunakan topologi jaringan grid dan topologi jaringan random

2.1 Wireless Sensor Network (WSN)

WSN adalah suatu jaringan nirkabel yang mempunyai kumpulan *node* sensor yang tersebar di suatu area tertentu. Setiap *node* mempunyai kemampuan pemrosesan, dapat memiliki sumber daya (baterai), beberapa jenis memori, mengakomodir berbagai sensor dan aktuator, serta *transceiver RF* (*Radio Frequency*). Contoh beberapa aplikasi yang dilakukan menggunakan jaringan sensor nirkabel adalah *area screening*, *node tracking scenarios*, panas dan kelembaban, *security monitoring*, sampai deteksi cahaya (Anggraini, 2012).

Sampai sekarang protokol 6LoWPAN dan Zigbee mendominasi dan bersaing pada pasar komunikasi nirkabel. Protokol 6LoWPAN dan Zigbee bisa dilakukan dalam komunikasi sensor nirkabel yang mempunyai daya rendah. *Node* pada Zigbee memungkinkan berada pada status rendah energi saat kondisi “tertidur”. Sedangkan 6LoWPAN beroperasi secara asinkronus, dengan algoritma routing yang tidak mengatur *nodes* dalam kondisi “tertidur” dan mengadopsi topologi *mesh*, karena dibutuhkan pendekatan *low-power listening* yang tidak hemat daya pada operasi ini (Tascano, 2012).

Faktor pembeda lain dari kedua protokol adalah *interoperability*. 6LoWPAN menawarkan interoperabilitas dengan IEEE 802.15.4 *physical link devices* serta perangkat bridges yang berada pada jaringan IP lain. Sehingga pengguna dapat berkomunikasi dengan perangkat lain dengan internet tanpa melakukan proses konversi *Zigbee to IP*.

Zigbee membutuhkan *gateway* yang digunakan untuk menghubungkan perangkat ke jaringan, sehingga dibutuhkan dua proses pengamanan yaitu satu dibagian internet dan satu dibagian jaringan. Proses ini menjadi pengaman dari serangan yang bisa mengancam integritas *end-to-end link*. Sedangkan 6LoWPAN tidak membutuhkan semua itu, sehingga header tambahan pun tidak diperlukan dan terdapat ruang lebih untuk besar *payload* (Tascano & Lo Bello, 2012).

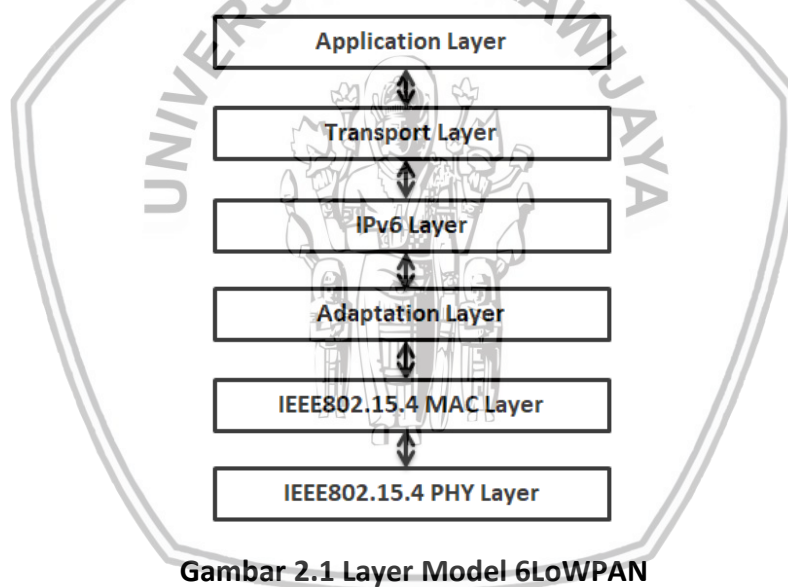
2.2 IEEE 802.15.4 dan 6LoWPAN

IEEE 802.15.4 merupakan standar *Media Access Control* (MAC) layer dan physical layer untuk perangkat yang menggunakan energi yang sedikit (beberapa baterai) dan perangkat berukuran kecil. IEEE 802.15.4 mempunyai karakter dan deskripsi sebagai berikut:

- *Low bandwidth*, dimana data rates yaitu 20 kbps, 40 kbps, 250 kbps, untuk masing-masing physical layer 868 Mhz, 915 Mhz, dan 2.4 Ghz.
- *Low power*, sebagian atau semuanya menggunakan baterai.
- *Low cost*, dimana perangkat menggunakan memori yang rendah serta mengadopsi kemampuan proses.
- Dimana jangkauan area sekitar 10m - 100m, biasanya area yang spesifik atau berupa ruang kecil.

- Perangkat yang terlibat merupakan *Reduced Function Device* (RFD) dan *Full Function Device* (FFD). FFD mempunyai peran sebagai router dan koordinator, router akan menentukan jalur dan koordinator akan membentuk jaringan, sementara RFD mempunyai peran sebagai *end device*, yaitu sensor yang berada dalam kondisi *sleep*
- Mempunyai dua mode MAC layer yaitu *beacon disabled* dan *beacon enabled* (Nurul, 2012).

6LoWPAN menggunakan IP yang digunakan untuk mengkoneksikan perangkat kecil berdaya rendah yang ada pada cakupan jaringan nirkabel. *Wireless Personal Area Network* (WPAN) merupakan jaringan untuk menghubungkan perangkat dalam area yang spesifik bersifat nirkabel. WPAN mempunyai jarak jangkauan yang pendek sekitar ± 100 meter. IPv6 digunakan dalam meningkatkan *interoperability* sehingga protokol 6LoWPAN dapat menyesuaikan diri pada revolusi jaringan yang IoT dan heterogeneous (Nurul, 2012). Layer model 6LoWPAN bisa dilihat pada Gambar 2.1



Gambar 2.1 Layer Model 6LoWPAN

Sumber : Shelby (2009)

Stack layer pada 6LoWPAN terdiri atas application layer, transport layer, network layer, adaptation layer, MAC layer, dan PHY layer. Pada hakikatnya 6LoWPAN memungkinkan device berstandar 802.15.4 dapat berkomunikasi satu sama lain secara nirkabel. Penjelasan stack layer dari 6LoWPAN adalah sebagai berikut:

- 6LoWPAN application layer

Pada layer ini menggunakan socket interface yang digunakan untuk aplikasi yang spesifik untuk menerima dan mengirim paket.

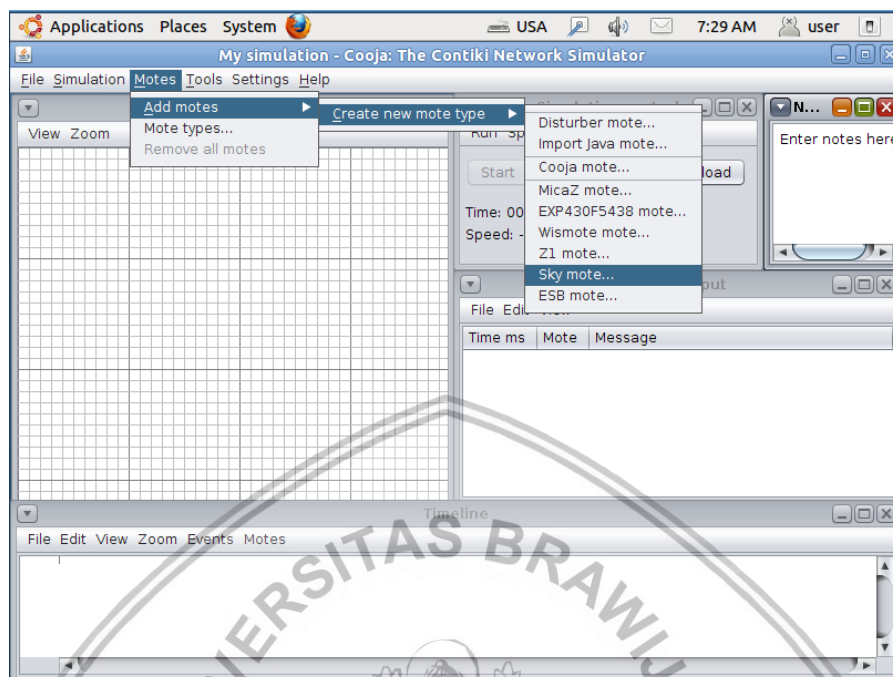
- 6LoWPAN transport layer
Pada transport layer mempunyai kesamaan pada layer OSI model, layer ini juga mempunyai tanggung jawab dalam process-to-process delivery.
- 6LoWPAN (IPv6) network layer
Pada layer ini mengakomodir interoperabilitas antar *node* sensor. Layer ini mempunyai fungsi utama yaitu pemetaan, routing protokol, dan pengalamatan.
- 6LoWPAN adaptation layer
6LoWPAN harus melakukan beberapa adaptasi dengan mengkomperasi header serta melakukan reassembly paket yang sebelumnya sudah difragmentasi yang digunakan 6LoWPAN mengadopsi format dari IPv6. 6LoWPAN mengatur routing pada layer ini.
- 6LoWPAN MAC layer
Jaringan IEEE 802.15.4 mempunyai empat standar untuk MAC layer frame, yaitu MAC data frame digunakan untuk transfer data, MAC beacon frame yang di-generate oleh koordinator untuk sinkronisasi, MAC command frame digunakan dalam MAC management entity, dan MAC Acknowledgement digunakan dalam meng-acknowledge paket yang berhasil terkirim.
- 6LoWPAN physical layer
Pada layer ini menyediakan dua layanan yaitu PHY *data services* yang mengatur penerimaan dan pengiriman paket data. PHY *management service* menyediakan akses ke dalam *management function* pada tiap layer serta menyimpan suatu informasi database *Personal Area Network* (PAN) yang saling terhubung.

2.3 Cooja Simulator

Cooja simulator adalah simulator jaringan WSN yang dikembangkan dan dibuat oleh Contiki OS project. Sudah banyak emulator/simulator yang bisa digunakan seperti TOSSIM dari TinyOS atau OpenSim dari OpenWSN. Namun cooja merupakan tool yang telah digunakan secara luas pada domain WSN dan telah dimasukkan didalam Avrora software dan MSPSim yang digunakan untuk melakukan *cycle-exact math emulation* (Roussel, 2016).

Beberapa komunitas riset pada umumnya menggunakan cooja untuk melakukan simulasi jaringan WSN, dari mulai yang luas sampai yang kecil serta melibatkan device sensor dan aktuator. Simulator ini digunakan untuk melakukan *virtual runs* pada jaringan sensor yang luas karena bisa secara signifikan menekan *cost*, adalah penggunaan perangkat yang banyak, serta biaya

yang besar dan waktu yang lama (Grag, 2015). Semnetara itu cooja simulator memiliki antarmuka pada Gambar 2.2.

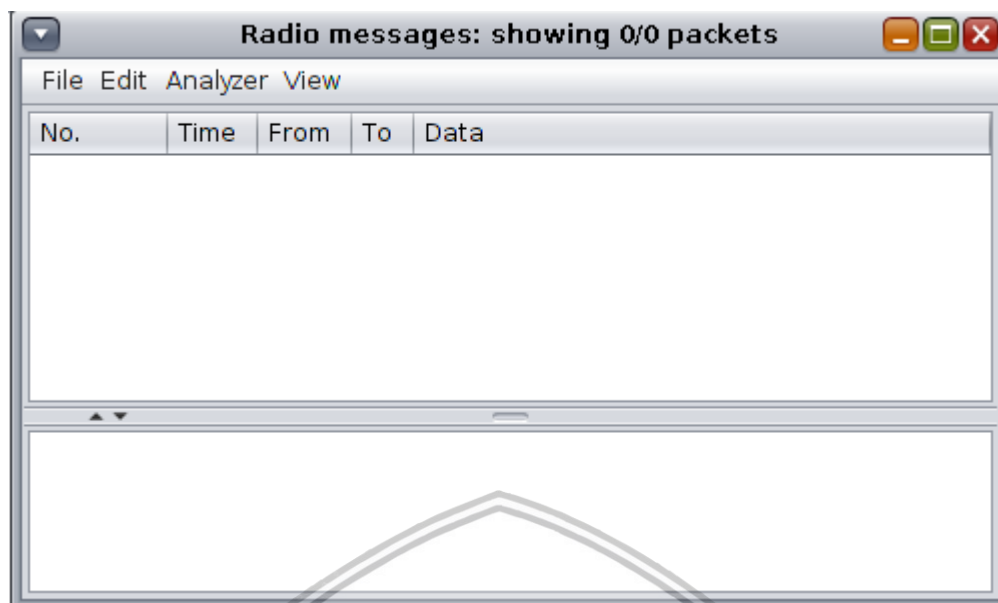


Gambar 2.2 Antar Muka Cooja Simulator

Node/motes yang digunakan adalah jenis skymote yang mempunyai *low power wireless sensor* yang dikembangkan dan didesain oleh University of California, Berkeley. Skymote mempunyai kemampuan yang digunakan untuk mengakomodir *high data rate sensor network* dengan membutuhkan energi kecil. Sensor ini juga digunakan untuk mengukur temperatur, kelembaban relatif, dan cahaya. Pada penelitian ini protokol yang diuji adalah 6LoWPAN dan routing protokol yang akan digunakan adalah RPL.

Skymotes dipilih karena saat process/firmware telah dipilih maka proses selanjutnya merupakan meng-create dan meng-compile motes yang nantinya akan terlihat langsung sebagai representasi motes yang sebenarnya pada simulator. Sehingga skymotes dipilih karena tipe motes ini mendukung protokol 6LoWPAN.

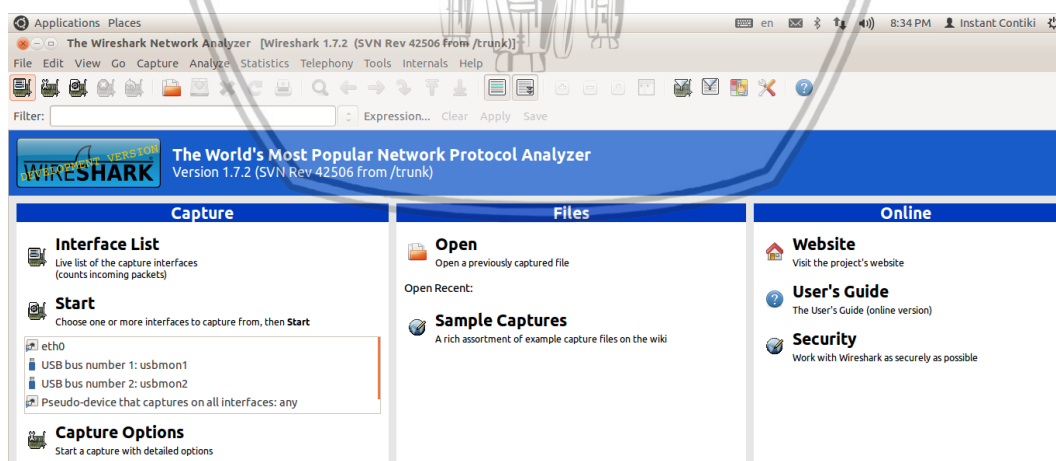
Cooja simulator mempunyai tools yang digunakan dalam penelitian ini yaitu *radio messages tools*. Tools ini dapat digunakan pengguna untuk melakukan tracking komunikasi data antar motes, asal dan tujuan paket tersebut, serta jumlah paket yang dikirim. Tools ini juga dapat mengetahui apakah motes yang digunakan mendukung protokol 6LoWPAN atau tidak. Tools ini pengguna juga bisa melakukan packet capture. Hasil dari packet capture tersebut berupa file .pcap yang bisa diakses pada folder home\contiki\tools\cooja\build. Antarmuka radio messages bisa dilihat pada Gambar 2.3



Gambar 2.3 Radio Messages Tools

2.4 Wireshark

Wireshark merupakan aplikasi *network packet analyzer* yang digunakan pengguna untuk menangkap suatu paket data pada suatu jaringan serta menampilkannya. Wireshark juga menampilkan apa yang terjadi pada jaringan internet. Pada situs resminya, bahwa mereka mengklaim wireshark merupakan *packet analyzer* berbasis *open source* terbaik sekarang ini. Tampilan wireshark bisa dilihat pada Gambar 2.4.



Gambar 2.4 Tampilan Antar Muka Wireshark

Dalam sisi fungsionalitasnya wireshark dapat dimanfaatkan untuk melakukan berbagai keperluan pada penelitian ini, diantaranya :

- Kebutuhan pengembang jaringan yang digunakan untuk meneliti penggunaan protokol

- Kebutuhan perseorangan yang digunakan untuk mempelajari protokol jaringan

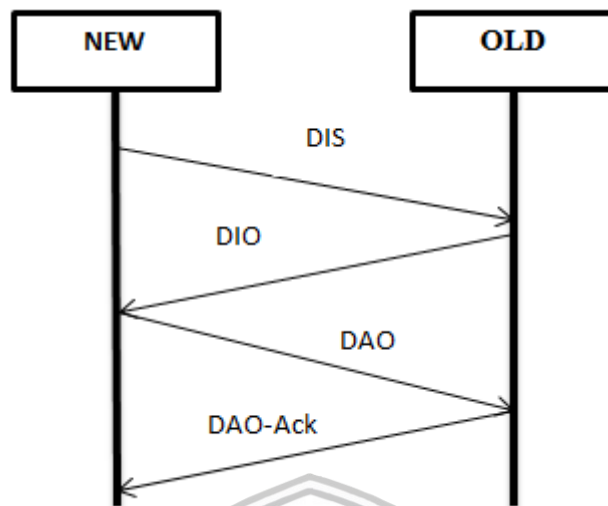
Pada penelitian ini fitur yang dimanfaatkan pada wireshark yaitu membuka file .pcap yang merupakan hasil simulasi dari jaringan WSN yang disimulasikan menggunakan aplikasi cooja simulator. File .pcap secara otomatis akan bisa dibuka menggunakan aplikasi wireshark. Untuk menganalisis protokol ini diperlukan data statistik dengan menggunakan indeks *quality of service*.

2.5 Routing Protocol for Low Power and Lossy Network (RPL)

Routing Protocol for Low Power and Lossy Network (RPL) merupakan distance vektor routing protokol yang berbasis pada DODAG's (Destination Oriented Directed Acyclic Graphs). DODAG sendiri merupakan bagian dari DAG (Directed Acyclic Graph) karena terfokus pada satu *node* tujuan. Basic terminology yang digunakan pada pembahasan RPL adalah sebagai berikut:

- DAG adalah graph yang tidak mempunyai cycle di dalamnya.
- Root adalah *node* tujuan yang akan dicari oleh *node* lain dalam DAG.
- DODAG merupakan DAG yang dimana setiap *node* menuju ke satu *node* tertentu sebagai tujuan.
- GOAL merupakan tujuan dari DODAG, misalnya ketika DODAG belum mencapai GOAL maka statusnya menjadi FLOATING dan jika DODAG mencapai GOAL nya maka statusnya menjadi GROUNDED.
- CHILD dan Parent adalah bisa asumsikan terdapat tiga *node* yaitu X, Y, Z, dimana anak panah dari Y dan Z mengarah ke X, maka X merupakan parent sementara itu Y dan Z merupakan Child. Child bisa mempunyai lebih dari satu Parent dan Parent juga bisa mempunyai lebih dari satu child.
- Storing adalah *node* yang mempunyai routing table, *node* yang berstatus storing bisa mengetahui cara mengirim paket dari satu *node* ke *node* yang lainnya.
- Non-storing adalah *node* yang dapat mengetahui parentnya saja.

Secara luas DODAG harus bisa menjaga keseragaman dengan semua *node* non-storing atau semua *node* storing, sedangkan root harus selalu dalam keadaan storing. Proses terbentuknya suatu DODAG melibatkan Control messages yang bisa dilihat pada Gambar 2.5



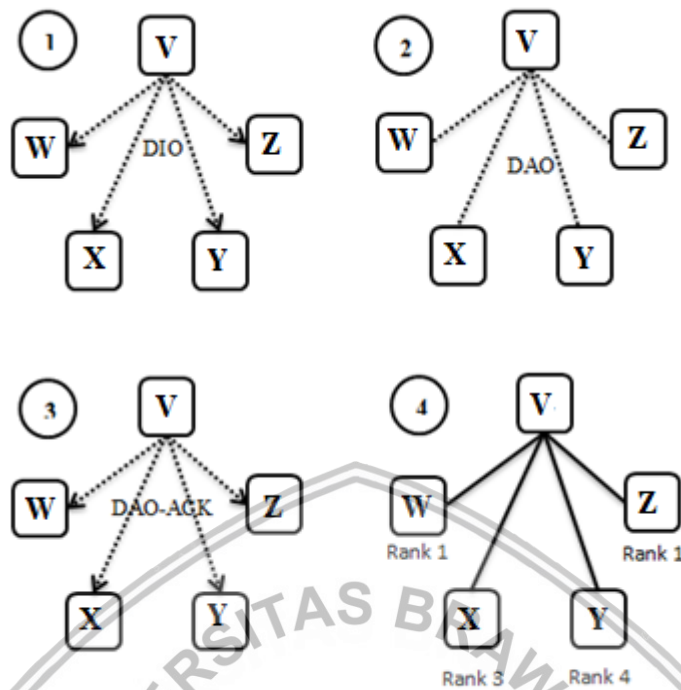
Gambar 2.5 Bagan Komunikasi Motes Baru Dan Motes Lama Pada DODAG

Sumber: Tsvetko (2011)

RPL menyediakan mekanisme untuk menyebarkan informasi diatas topologi jaringan yang bersifat dinamis. Pesan-pesan kontrol yang digunakan dalam RPL meliputi: DIO (*DODAG Information Object*), DAO (*Destination Advertisement Object*), DIS (*DODAG Information Solicitation*). NEW dan OLD merupakan sebuah istilah dari node baru yang akan bergabung dalam DODAG dan node lama yang sudah bergabung dalam DODAG. Jika ada sebuah node yang ingin bergabung pada sebuah DODAG, maka node tersebut akan mengirimkan pesan DIS, dan node terdekat yang sudah berada dalam DODAG akan membalas dengan mengirimkan pesan DIO. Setelah mendapat pesan DIO, node baru tersebut mengirimkan pesan DAO, dan akhirnya node lama meresponnya dengan mengirimkan pesan DAO-ack, seperti terlihat pada Gambar 2.6

- DIS adalah pesan yang akan dikirimkan oleh *node* untuk mencari tahu apakah ada DODAG disekitarnya bagi node yang belum bergabung dalam DODAG.
- DIO merupakan *control messages* yang dikirimkan oleh root kesemua *node* dalam DODAG. *Control messages* tersebut diberikan sebuah informasi mengenai status dari DODAG tersebut, *floating* atau *grounded*, *non-storing* atau *storing*. Pesan ini berisikan kurang lebih seperti “jika ada *node* yang ingin bergabung, tolong kirimkan pesan balasan”.
- DAO merupakan sebuah pesan meminta bergabung kepada root atau parent-nya yang dikirimkan *node* sebagai respon dari DIO.
- DAO-ACK adalah respon pemberian izin untuk bergabung terhadap pesan DAO.

Proses terbentuknya jaringan routing RPL bisa dilihat pada Gambar 2.6:



Gambar 2.6 Ilustrasi Terbentuknya DODAG

Sumber: Tsvetko (2011)

1. Misalnya terdapat 5 node V, W, X, Y, Z dan akan membuat DODAG
2. V me-multicast DIO
3. Node lainnya yaitu W, X, Y, Z akan menerima DIO dan akan mencoba untuk bergabung, selain itu W, X, Y, Z akan mengetahui jarak mereka ke A misalkan yaitu 1, 1, 3, 4.
4. W, X, Y, Z kemudian mengirimkan DAO ke V
5. V meresponnya dengan DAO-ACK
6. Karena W dan X merupakan node terdekat dengan root pada rank 1 lalu mengirimkan DIO
7. Y menerima DIO dari W dan X, serta mengetahui jarak dari mereka yaitu 1 dan 2
8. Z juga menerima DIO dari W dan X, serta mengetahui jarak dari mereka yaitu 2 dan 1
9. Karena Y lebih dekat ke W dan Z lebih dekat ke X, maka Y mengirim DAO ke W dan Z mengirim DAO ke X. W mengirim DAO-ACK ke Y dan X mengirim DAO-ACK ke Z.
10. Dengan ini informasi DODAG telah tersusun.

2.6 Quality of Service (QoS)

QoS dalam sebuah telekomunikasi yang digunakan pengguna sebagai kebutuhan yang disediakan oleh jaringan (Fabricio, 2009). Pengguna akan menggunakan kinerja yang dibutuhkan pada suatu jaringan dalam bentuk parameter QoS seperti *throughput* atau *delay*. QoS memiliki parameter berbeda pada setiap model.

- **Throughput**

Throughput merupakan jumlah data yang diproses dalam satuan waktu tertentu atau jumlah data yang diterima atau dikirim oleh suatu jaringan. *Throughput* memiliki satuan khusus yaitu bit/s atau bps (Fabricio, 2009). Rumus *Throughput* dapat dihitung sebagai berikut :

$$\text{Throughput} = \frac{\text{Jumlah data yang dikirim (dalam satuan bps)}}{\text{waktu (dalam satuan second)}}$$

Sumber: (Ross dan Kurose, 2013)

- **Delay**

Delay merupakan waktu yang dibutuhkan untuk mengirimkan data dari node asal sampai node tujuan. Waktu *delay* bisa meningkat bila mengalami antrian yang panjang atau melewati rute yang lebih jauh pada suatu jaringan. *Delay* bisa dihitung round-trip (waktu yang dibutuhkan dari asal paket ke tujuan lalu kembali lagi ke asal) atau secara langsung (total waktu dari paket dikirim sampai paket diterima) (Fabricio, 2009). Rata-rata *Delay* dapat dihitung dengan rumus berikut ini :

$$\text{Average Delay} = \frac{\text{Total delay}}{\text{total paket yang diterima}}$$

Total delay =

(Waktu Pengirim 1 – Waktu Penerima 1) + (Waktu Pengirim 2 – Waktu Penerima 2) + ... (Waktu Pengirim n – Waktu Penerima n) ...

- **Jitter**

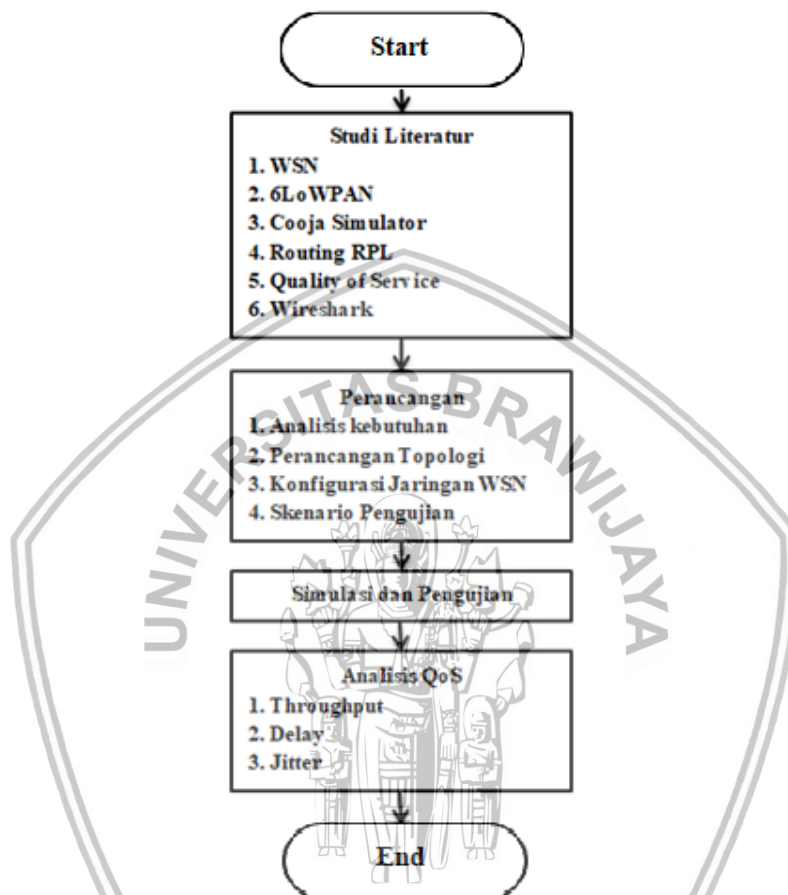
Jitter merupakan variasi *delay* yang berhubungan dengan kedatangan paket. *Jitter* diakibatkan oleh reassemble data diakhir pengiriman dan panjang queue dalam satu pengolahan data. Maka semakin besar trafik karena variasi *delay* akan membuat terjadinya tumpukan antar paket juga akan tinggi sehingga nilai QoS menurun dan nilai *jitter* akan makin besar (Fabricio, 2009). Rumus *jitter* dapat dihitung sebagai berikut :

$$\text{Jitter} = \frac{\text{Total variasi delay}}{\text{total paket yang diterima} - 1}$$

Sumber : (Ross dan Kurose, 2013).

BAB 3 METODOLOGI

Pada sub bab ini menjelaskan tahapan metodologi yang digunakan dalam penelitian. Tahapan-tahapan penelitian digambarkan dalam diagram alir pada Gambar 3.1



Gambar 3.1 Metodologi Penelitian

Berdasarkan Gambar 3.1, maka dapat dijelaskan tahapan penelitian diagram alir adalah sebagai berikut :

1. Studi literatur sesuai dengan topik permasalahan yang dilakukan dalam penelitian ini.
2. Perancangan kebutuhan dan simulasi jaringan WSN yang akan digunakan dalam pengujian nantinya.
3. Membuat simulasi jaringan sesuai dengan perancangan yang dilakukan sebelumnya dan melakukan pengujian.
4. Analisis hasil pengujian dengan parameter uji antara lain yaitu *Throughput*, *Delay*, dan *Jitter*.
5. Penarikan kesimpulan dari hasil penelitian serta memberikan saran-saran untuk penelitian berikutnya.

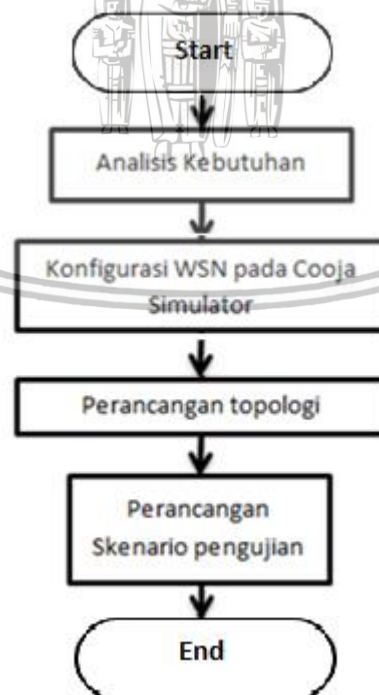
3.1 Studi Literatur

Pada bagian studi literatur ini digunakan untuk mendalami konsep yang terkait dengan simulasi protokol 6LoWPAN pada jaringan sensor nirkabel dengan menggunakan cooja simulator. Studi literatur ini mengenai karakteristik, parameter, dan teori pendukung lain yang menunjang dalam penelitian ini. Teori pendukung tersebut adalah :

1. Wireless Sensor Network (WSN)
2. IEEE 802.15.4 dan 6LoWPAN
3. Cooja Simulator
4. Wireshark
5. Routing Protocol for Low Power and Lossy Network
6. Quality of Service
 - a. *Throughput*
 - b. *Delay*
 - c. *Jitter*

3.2 Perancangan

Perancangan dibuat dengan menerapkan protokol 6LoWPAN pada simulasi jaringan WSN dengan menggunakan topologi grid dan random pada cooja simulator. Sistem perancangan alur penelitian bisa dilihat pada Gambar 3.2



Gambar 3.2 Perancangan Sistem

Tahapan perancangan pada Gambar 3.2 dapat dijelaskan sebagai berikut :

1. Analisis kebutuhan dari kebutuhan perangkat keras dan perangkat lunak,
2. Konfigurasi WSN pada cooja simulator dengan memilih motes yang digunakan,
3. Konfigurasi motes yang telah dipilih lalu merancang topologi,
4. Perancangan skenario pengujian dengan parameter yang diuji.
5. Selesai.

3.3 Simulasi dan Pengujian

Simulasi akan mengacu pada perancangan sistem yang meliputi sebagai berikut:

1. Mengkonfigurasi cooja simulator
2. Merancang jaringan sensor nirkabel pada cooja simulator
3. Merancang dan menjalankan simulasi motes dengan protokol 6LoWPAN pada jaringan WSN dengan topologi grid dan topologi random menggunakan cooja simulator
4. Menganalisis kinerja jaringan dengan wireshark dan mote output secara manual

3.4 Analisis

Analisis dilakukan menggunakan metode analisis kualitatif. Dimana setelah semua pengujian dilakukan secara menyeluruh, dengan membandingkan hasil yang diperoleh kedalam bentuk grafik atau tabel. Setiap grafik dan tabel akan mempresentasikan hasil pengujian secara menyeluruh yang mencakup tiga parameter QoS yang diuji yaitu *Throughput*, *Delay*, dan *Jitter*.

3.5 Pengambilan Kesimpulan

Kesimpulan disusun berdasarkan hasil dan analisis pengujian terhadap penelitian ini. Pengambilan kesimpulan dibuat setelah perancangan, implementasi, pengujian dan analisis dilakukan. Isi kesimpulan diharapkan bisa menjadi acuan untuk simulasi 6LoWPAN pada jaringan WSN. Akhir dari penulisan ini dibuat saran untuk penyempurnaan dan memperbaiki kesalahan terhadap skripsi yang dilakukan ini.

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Pada sub bab ini menjelaskan perancangan dan mengimplementasikan skenario pengujian dan hasil sebagian pengujian simulasi yang telah dijalankan dengan menggunakan cooja simulator. Implementasi skenario pengujian mencakup pengujian simulasi dengan motes yang digunakan dan mendukung 6LoWPAN, menerapkan *Routing Protocol for Lower Power on Lossy Network* dan penggunaan aplikasi *wireshark* dan hasil dari Mote Output untuk menampilkan akumulasi paket data yang telah dikirim dan diterima oleh setiap nodes sesuai parameter yang diuji.

4.1 Perancangan

Perancangan digunakan untuk menjelaskan rancangan konfigurasi motes yang digunakan berbasis protokol 6LoWPAN pada jaringan WSN. Ada hal yang akan dibahas dari perancangan simulasi ini yaitu Analisis kebutuhan, Perancangan topologi, Perancangan skenario pengujian. Ada beberapa hal yang diperlukan untuk menerapkan sistem adalah mempersiapkan perangkat keras dan perangkat lunak.

4.1.1 Analisis Kebutuhan

Pada analisis kebutuhan membahas tentang bagaimana menyelesaikan masalah pada penelitian ini yang akan dibangun, dalam hal ini simulasi jaringan WSN dengan protokol 6LoWPAN pada topologi jaringan grid dan topologi jaringan random menggunakan cooja simulator. Pada bagian ini dijelaskan cara menggunakan cooja yang digunakan untuk membangun lingkungan simulasi jaringan WSN, perancangan topologi dan skenario pengujian. Semuanya akan menjadi analisis kebutuhan untuk menjawab sebuah rumusan masalah yang akan dibahas di penelitian ini.

- **Perangkat Keras Yang Digunakan**

Pada penelitian ini digunakan satu buah laptop untuk mengkonfigurasi simulasi WSN serta simulasi motes berbasis protokol 6LoWPAN. Perangkat keras yang digunakan bisa dilihat pada Tabel 4.1 berikut,

Tabel 4.1 Perangkat Keras yang Digunakan

No	Perangkat Keras	Fungsi
1	Laptop Lenovo	Digunakan untuk menkonfigurasi dan menjalankan simulasi protokol 6LoWPAN, serta menulis dokumen

- **Perangkat Lunak Yang Digunakan**

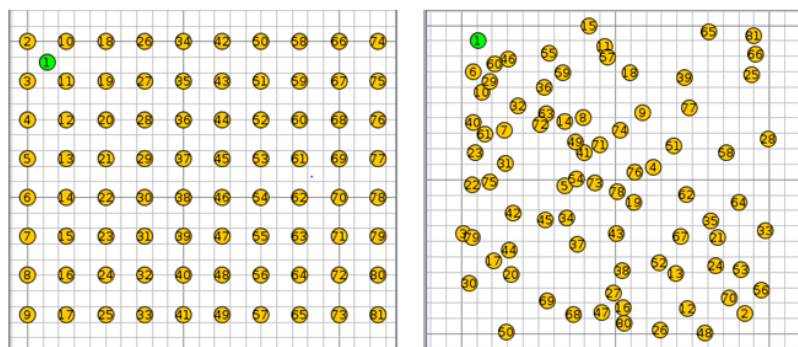
Pada penelitian ini menggunakan beberapa perangkat lunak untuk merancang sebuah jaringan sensor nirkabel untuk mengimplementasikan 6LoWPAN. Perangkat lunak yang akan digunakan dapat dilihat pada Tabel 4.2 sebagai berikut :

Tabel 4.2 Perangkat Lunak yang Digunakan

No	Perangkat Lunak	Fungsi
1	VMware Workstation v12.5.9	Membuat virtual machine untuk menjalankan instan contiki
2	Cooja Simulator v2.7	Simulator jaringan WSN
3	Wireshark	Digunakan untuk menampilkan parameter QoS yang dianalisis
4	MS Excel 2010	Digunakan untuk menghitung parameter QoS yang dianalisis
5	Windows	Sebagai Host
6	Ubuntu	Sebagai Server

4.1.2 Perancangan Topologi

Perancangan simulasi jaringan WSN yang digunakan untuk mengetahui kinerja protokol 6LoWPAN dilakukan menggunakan cooja simulator. Pada simulasi jaringan ini menggunakan topologi yang berbeda yaitu topologi grid dan topologi random. Masing-masing topologi akan diberi jumlah *node* yang sama untuk mengetahui kinerja dari masing-masing topologi tersebut. Jumlah *nodes* yang akan digunakan yaitu; 10, 20, 40, dan 80 karena untuk membedakan jarak antar node dengan jumlah yang berbeda, serta batasan luas area yang digunakan dalam penelitian ini memiliki jangkauan pendek sekitar ± 100 meter. *Node* yang digunakan merupakan node tidak berpindah-pindah. Pada pembangunan topologi ini fokus pada *Routing* RPL yang digunakan memonitoring skema salah satu QoS yaitu *Delay*. Referensi topologi grid dan topologi random yang digunakan dapat dilihat pada Gambar 4.1.



Gambar 4.1 Topologi Grid dan Topologi Random

Pada Gambar 4.1 node yang digunakan ada dua jenis yaitu, node yang berwarna hijau adalah node receiver, sedangkan node yang berwarna kuning adalah node sender. Ruang lingkup simulasi Cooja Simulator tidak menyertakan fasilitas atau tools yang memungkinkan pengguna menentukan jarak antar motes secara otomatis, akan tetapi bisa diketahui bahwa penempatan motes pada simulasi ini berbasiskan grid 10m x 10m.

4.1.3 Perancangan Skenario Pengujian

Perancangan skenario pengujian dilakukan untuk menentukan urutan serta implementasi pengujian sehingga data yang didapatkan untuk di analisis secara maksimal. Pada penelitian ini pengujian dilakukan dengan melibatkan 10, 20, 40, dan 80 *node* pada dua topologi yang berbeda dengan menggunakan luas area 100x100 meter karena untuk mengoptimalkan jarak jangkauan antar node yang memiliki jarak yang kecil. Simulasi ini dijalankan selama 500 detik pada *node* yang berbeda-beda karena 2 menit awal digunakan untuk menginisiasi alamat IPv6 pada semua node, sedangkan pada interval 2 - 10 menit merupakan menit yang optimal untuk melakukan simulasi. Maka dari itu memilih 500 detik karena sebagai standar minimum untuk melakukan simulasi. Parameter simulasi yang digunakan dapat dilihat pada Tabel 4.3

Tabel 4.3 Parameter Simulasi

Network Simulator	Cooja Simulator
Protokol <i>Routing</i>	RPL
Mobility	Tidak berpindah-pindah
Lama waktu pengujian	500 detik
Luas Area	100 m x 100 m
Wifi range	50 m
Data rate	20 Kb
Ukuran data	128 byte

Hasil dari simulasi kemudian akan ditangkap menjadi file .pcap dan file .txt yang kemudian dianalisis menggunakan wireshark dan MS. Excel untuk memperoleh parameter QoS yang dianalisis yaitu *Throughput*, *Delay*, dan *Jitter* untuk setiap paket yang dikirimkan.

4.2 Implementasi

Implementasi dilakukan terhadap seluruh lingkungan pengujian yang dibutuhkan sesuai dengan rancangan yang dibuat sebelumnya.

4.2.1 Membuat Topologi dan Menjalankan Simulasi

Topologi yang akan digunakan dalam penelitian ini sesuai dengan yang dibahas sebelumnya. Topologi yang digunakan ada 2 yaitu, topologi random dan topologi grid. Pada masing-masing topologi melibatkan 10, 20, 40, dan 80 node untuk dilakukan sebuah pengujian. Perancangan topologi grid dan topologi random dapat dilihat pada Gambar 4.2, Gambar 4.3, Gambar 4.4, dan Gambar 4.5 berikut ini;

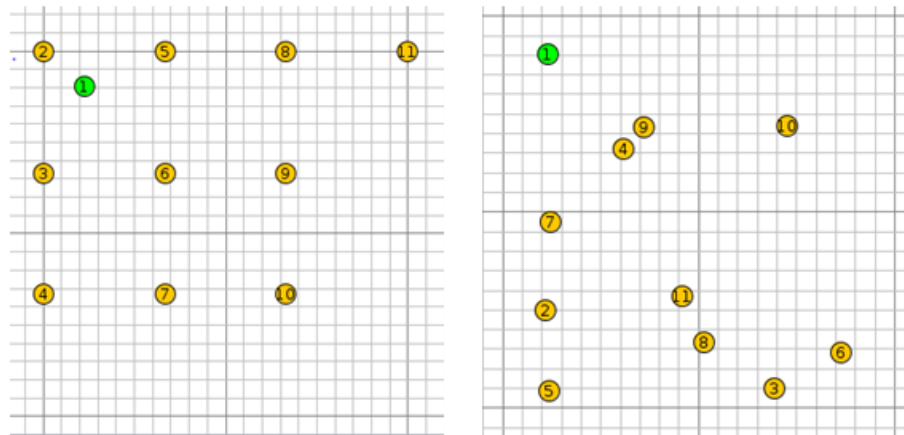
a. Skenario 10 node

Dalam menjalankan implementasi skenario ini, dibutuhkan 1 node receiver dan 10 node sender. 10 node sender tersebut akan mengirimkan paket berulang-ulang ke node receiver selama simulasi ini yang dijalankan yaitu 500 detik. Dalam setiap node mempunyai alamat IP seperti pada Tabel 4.4

Tabel 4.4 Konfigurasi Topologi Skenario 10 Node

#destination	aaaa::212:740b:b:b0b
#source	aaaa::212:7401:1:101 aaaa::212:7402:2:202 aaaa::212:7403:3:303 aaaa::212:7404:4:404 aaaa::212:7405:5:505 aaaa::212:7406:6:606 aaaa::212:7407:7:707 aaaa::212:7408:8:808 aaaa::212:7409:9:909 aaaa::212:740a:a:a0a

Berikut adalah tampilan rancangan simulasi pada cooja simulator sebagaimana pada Gambar 4.2. node hijau sebagai receiver dan node kuning sebagai sender.



Gambar 4.2 Topologi Grid dan Topologi Random Menggunakan 10 node

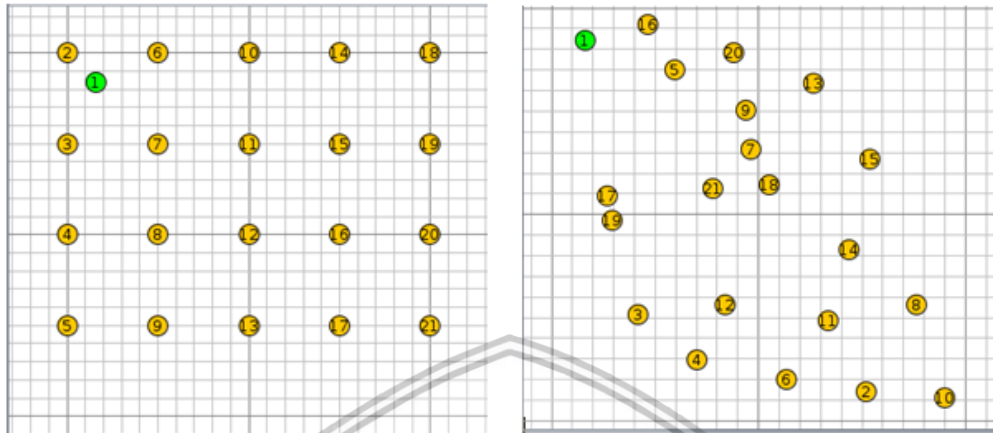
b. Skenario 20 node

Dalam menjalankan implementasi skenario ini, dibutuhkan 1 node receiver dan 20 node sender. 20 node sender tersebut akan mengirimkan paket berulang-ulang ke node receiver selama simulasi ini yang dijalankan yaitu 500 detik. Dalam setiap node mempunyai alamat IP seperti pada Tabel 4.5

Tabel 4.5 Konfigurasi Topologi Skenario 20 Node

#destination	aaaa::212:7415:15:1515
#source	aaaa::212:7401:1:101 aaaa::212:7402:2:202 aaaa::212:7403:3:303 aaaa::212:7404:4:404 aaaa::212:7405:5:505 aaaa::212:7406:6:606 aaaa::212:7407:7:707 aaaa::212:7408:8:808 aaaa::212:7409:9:909 aaaa::212:740a:a:a0a aaaa::212:740b:b:b0b aaaa::212:740c:c:c0c aaaa::212:740d:d:d0d aaaa::212:740e:e:e0e aaaa::212:740f:f:f0f aaaa::212:7410:10:1010 aaaa::212:7411:11:1111 aaaa::212:7412:12:1212 aaaa::212:7413:13:1313 aaaa::212:7414:14:1414

Berikut adalah tampilan rancangan simulasi pada cooja simulator sebagaimana pada Gambar 4.3. node hijau sebagai receiver dan node kuning sebagai sender.



Gambar 4.3 Topologi Grid dan Topologi Random Menggunakan 20 node

c. Skenario 40 node

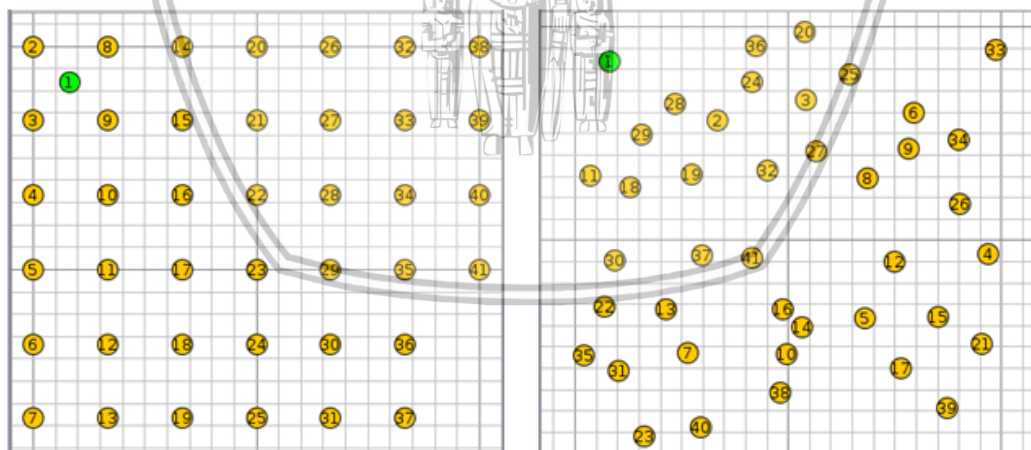
Dalam menjalankan implementasi skenario ini, dibutuhkan 1 node receiver dan 40 node sender. 40 node sender tersebut akan mengirimkan paket berulang-ulang ke node receiver selama simulasi ini yang dijalankan yaitu 500 detik. Dalam setiap node mempunyai alamat IP seperti pada Tabel 4.6

Tabel 4.6 Konfigurasi Topologi Skenario 40 Node

#destination	aaaa::212:7429:29:2929
#source	aaaa::212:7401:1:101 aaaa::212:7402:2:202 aaaa::212:7403:3:303 aaaa::212:7404:4:404 aaaa::212:7405:5:505 aaaa::212:7406:6:606 aaaa::212:7407:7:707 aaaa::212:7408:8:808 aaaa::212:7409:9:909 aaaa::212:740a:a:a0a aaaa::212:740b:b:b0b aaaa::212:740c:c:c0c aaaa::212:740d:d:d0d aaaa::212:740e:e:e0e aaaa::212:740f:f:f0f aaaa::212:7410:10:1010 aaaa::212:7411:11:1111 aaaa::212:7412:12:1212 aaaa::212:7413:13:1313

aaaa::212:7414:14:1414
aaaa::212:7415:15:1515
aaaa::212:7416:16:1616
aaaa::212:7417:17:1717
aaaa::212:7418:18:1818
aaaa::212:7419:19:1919
aaaa::212:741a:1a:1a1a
aaaa::212:741b:1b:1b1b
aaaa::212:741c:1c:1c1c
aaaa::212:741d:1d:1d1d
aaaa::212:741e:1e:1e1e
aaaa::212:741f:1f:1f1f
aaaa::212:7420:20:2020
aaaa::212:7421:21:2121
aaaa::212:7422:22:2222
aaaa::212:7423:23:2323
aaaa::212:7424:24:2424
aaaa::212:7425:25:2525
aaaa::212:7426:26:2626
aaaa::212:7427:27:2727
aaaa::212:7428:28:2828

Berikut adalah tampilan rancangan simulasi pada cooja simulator sebagaimana pada Gambar 4.4. node hijau sebagai receiver dan node kuning sebagai sender.



Gambar 4.4 Topologi Grid dan Topologi Random Menggunakan 40 node

d. Skenario 80 node

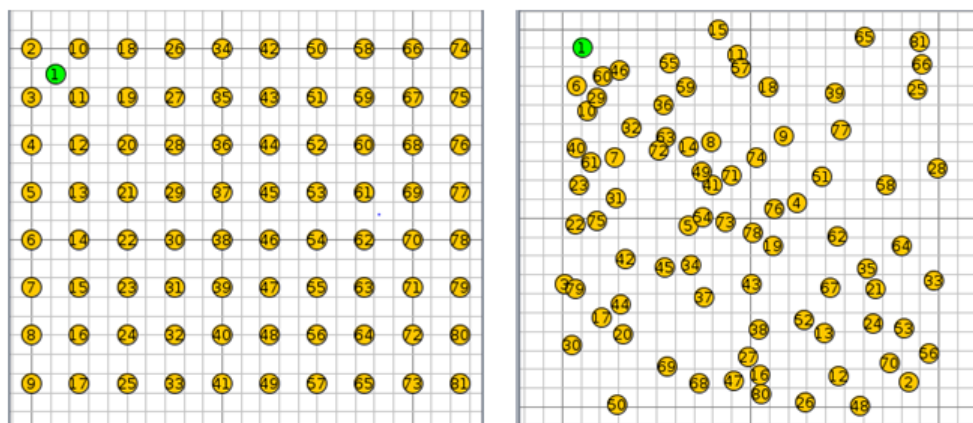
Dalam menjalankan implementasi skenario ini, dibutuhkan 1 node receiver dan 80 node sender. 80 node sender tersebut akan mengirimkan paket berulang-ulang ke node receiver selama simulasi ini yang dijalankan yaitu 500 detik. Dalam setiap node mempunyai alamat IP seperti pada Tabel 4.7

Tabel 4.7 Konfigurasi Topologi Skenario 80 Node

#destination	aaaa::212:7451:51:5151
#source	aaaa::212:7401:1:101 aaaa::212:7402:2:202 aaaa::212:7403:3:303 aaaa::212:7404:4:404 aaaa::212:7405:5:505 aaaa::212:7406:6:606 aaaa::212:7407:7:707 aaaa::212:7408:8:808 aaaa::212:7409:9:909 aaaa::212:740a:a:a0a aaaa::212:740b:b:b0b aaaa::212:740c:c:c0c aaaa::212:740d:d:d0d aaaa::212:740e:e:e0e aaaa::212:740f:f:f0f aaaa::212:7410:10:1010 aaaa::212:7411:11:1111 aaaa::212:7412:12:1212 aaaa::212:7413:13:1313 aaaa::212:7414:14:1414 aaaa::212:7415:15:1515 aaaa::212:7416:16:1616 aaaa::212:7417:17:1717 aaaa::212:7418:18:1818 aaaa::212:7419:19:1919 aaaa::212:741a:1a:1a1a aaaa::212:741b:1b:1b1b aaaa::212:741c:1c:1c1c aaaa::212:741d:1d:1d1d aaaa::212:741e:1e:1e1e aaaa::212:741f:1f:1f1f aaaa::212:7420:20:2020 aaaa::212:7421:21:2121 aaaa::212:7422:22:2222 aaaa::212:7423:23:2323 aaaa::212:7424:24:2424 aaaa::212:7425:25:2525 aaaa::212:7426:26:2626 aaaa::212:7427:27:2727 aaaa::212:7428:28:2828 aaaa::212:7429:29:2929 aaaa::212:742a:2a:2a2a

	aaaa::212:742b:2b:2b2b
	aaaa::212:742c:2c:2c2c
	aaaa::212:742d:2d:2d2d
	aaaa::212:742e:2e:2e2e
	aaaa::212:742f:2f:2f2f
	aaaa::212:7430:30:3030
	aaaa::212:7431:31:3131
	aaaa::212:7432:32:3232
	aaaa::212:7433:33:3333
	aaaa::212:7434:34:3434
	aaaa::212:7435:35:3535
	aaaa::212:7436:36:3636
	aaaa::212:7437:37:3737
	aaaa::212:7438:38:3838
	aaaa::212:7439:39:3939
	aaaa::212:743a:3a:3a3a
	aaaa::212:743b:3b:3b3b
	aaaa::212:743c:3c:3c3c
	aaaa::212:743d:3d:3d3d
	aaaa::212:743e:3e:3e3e
	aaaa::212:743f:3f:3f3f
	aaaa::212:7440:40:4040
	aaaa::212:7441:41:4141
	aaaa::212:7442:42:4242
	aaaa::212:7443:43:4343
	aaaa::212:7444:44:4444
	aaaa::212:7445:45:4545
	aaaa::212:7446:46:4646
	aaaa::212:7447:47:4747
	aaaa::212:7448:48:4848
	aaaa::212:7449:49:4949
	aaaa::212:744a:4a:4a4a
	aaaa::212:744b:4b:4b4b
	aaaa::212:744c:4c:4c4c
	aaaa::212:744d:4d:4d4d
	aaaa::212:744e:4e:4e4e
	aaaa::212:744f:4f:4f4f
	aaaa::212:7450:50:5050

Berikut adalah tampilan rancangan simulasi pada cooja simulator sebagaimana pada Gambar 4.5. node hijau sebagai receiver dan node kuning sebagai sender.



Gambar 4.5 Topologi Grid dan Topologi Random Menggunakan 80 node

4.2.2 Mengelola Hasil Simulasi Cooja Simulator

Dalam penelitian ini aplikasi wireshark digunakan untuk mengolah data file packet capture yaitu (.pcap) dari cooja simulator yang sebelumnya disimulasikan untuk mengetahui nilai *Throughput* dari masing-masing topologi grid dan topologi random dengan jumlah node yang berbeda-beda. Parameter *Throughput* dapat dilihat pada *capture file properties* seperti pada topologi grid menggunakan 10 node di Gambar 4.6 berikut

Statistics			
Measurement	Captured	Displayed	Marked
Packets	4300	212 (4.9%)	—
Time span, s	312.596	227.428	—
Average pps	13.8	0.9	—
Average packet size, B	78	93	—
Bytes	334312	19734 (5.9%)	0
Average bytes/s	1069	86	—
Average bits/s	8555	694 Throughput	—

Gambar 4.6 Capture File Properties Topologi Grid Menggunakan 10 Node

Parameter *delay* dapat dilihat dengan menampilkan hasil dari data capture mote output (.txt), seperti dapat dilihat pada Gambar 4.7

```
loglistener.txt *
03:09.939 ID:6 Sending unicast to aaaa::212:740b:b:b0b
03:10.121 ID:11 Data received from aaaa::212:7406:6:606 on port 1234 from port 1234 with length 10: 'Message 2'
03:25.601 ID:1 Sending unicast to aaaa::212:740b:b:b0b
03:25.748 ID:11 Data received from aaaa::212:7401:1:101 on port 1234 from port 1234 with length 10: 'Message 2'
03:28.409 ID:8 Sending unicast to aaaa::212:740b:b:b0b
03:28.622 ID:11 Data received from aaaa::212:7408:8:808 on port 1234 from port 1234 with length 10: 'Message 2'
03:36.105 ID:10 Sending unicast to aaaa::212:740b:b:b0b
03:36.621 ID:11 Data received from aaaa::212:740a:a:a0a on port 1234 from port 1234 with length 10: 'Message 2'
03:37.490 ID:9 Sending unicast to aaaa::212:740b:b:b0b
03:37.745 ID:11 Data received from aaaa::212:7409:9:909 on port 1234 from port 1234 with length 10: 'Message 2'
03:41.803 ID:4 Sending unicast to aaaa::212:740b:b:b0b
03:41.870 ID:11 Data received from aaaa::212:7404:4:404 on port 1234 from port 1234 with length 10: 'Message 2'
03:43.010 ID:3 Sending unicast to aaaa::212:740b:b:b0b
03:43.245 ID:11 Data received from aaaa::212:7403:3:303 on port 1234 from port 1234 with length 10: 'Message 2'
03:48.293 ID:7 Sending unicast to aaaa::212:740b:b:b0b
03:48.497 ID:11 Data received from aaaa::212:7407:7:707 on port 1234 from port 1234 with length 10: 'Message 2'
```

Gambar 4.7 Capture File Mote Output

Kemudian hasil *capture* diolah secara manual di MS. Excel untuk menghitung rata-rata *delay* dengan rumus average. Hasil dari rata-rata *delay* bisa dilihat pada Gambar 4.8

	A	B	C
1	Delay	Total Delay	Average Delay
2	0,087	15,321	0,072268868
3	0,041		
4	0,218		
5	0,227		
6	0,291		
7	0,126		
8	0,261		
9	0,187		
10	0,159		
11	0,229		
12	0,118		
13	0,167		
14	0,165		
15	0,158		
16	0,174		
17	0,061		
18	0,213		

Gambar 4.8 Rata-Rata Delay

Kolom *delay* pada Gambar 4.8 juga bisa digunakan untuk menghitung variasi *delay* yang jika dibagikan dengan jumlah paket akan menghasilkan *jitter*. Hasil untuk memperoleh parameter *jitter* bisa dilihat pada Gambar 4.9

E	F	G	H	I	J	K
Delay (n)	Delay(n-1)		Jitter	Absolut Jitter	Total Variasi Delay	Average Jitter
0,087	0,041		-0,046	0,046	7,946	0,037481132
0,041	0,218		0,177	0,177		
0,218	0,227		0,009	0,009		
0,227	0,291		0,064	0,064		
0,291	0,126		-0,165	0,165		
0,126	0,261		0,135	0,135		
0,261	0,187		-0,074	0,074		
0,187	0,159		-0,028	0,028		
0,159	0,229		0,07	0,07		
0,229	0,118		-0,111	0,111		
0,118	0,167		0,049	0,049		
0,167	0,165		-0,002	0,002		
0,165	0,158		-0,007	0,007		
0,158	0,174		0,016	0,016		
0,174	0,061		-0,113	0,113		
0,061	0,213		0,152	0,152		
0,213	0,066		-0,147	0,147		
0,066	0,258		0,192	0,192		
0,258	0,246		-0,012	0,012		
0,246	0,108		-0,138	0,138		

Gambar 4.9 Total Jitter

BAB 5 HASIL DAN PEMBAHASAN

Pada bab ini analisis seluruh hasil pengujian yang telah dilakukan pada bab sebelumnya. Bab ini dibagi menjadi tiga sub bab hasil dan analisis sesuai dengan jumlah parameter QoS yang diuji yaitu *throughput*, *delay*, dan *jitter* dari masing-masing topologi dengan jumlah node berbeda-beda yang akan dibandingkan.

5.1 Hasil Pengujian

Pengujian ini digunakan untuk mengetahui parameter *Throughput*, *delay* dan *jitter* dari simulasi WSN dengan protocol 6LoWPAN. Pengujian ini dilakukan dengan menggunakan 10, 20, 40 dan 80 node dalam topologi grid dan topologi random. Simulasi dijalankan selama 500 detik dengan node sender akan mengirimkan paket berulang-ulang ke node receiver, selama simulasi berlangsung radio messages tools akan melakukan packet capture. Hasil *packet capture* berupa file .pcap ini kemudian di analisis dengan wireshark.

5.1.1 Hasil Throughput

Pada sub bab ini ditampilkan hasil pengujian terhadap parameter *throughput*. Pengujian ini melibatkan 10, 20, 40, dan 80 node pada topologi grid dan topologi random dengan dilakukan selama 500 detik. Selama pengujian berlangsung radio messages tools akan melakukan packet capture yang menghasilkan file .pcap untuk kemudian di analisis menggunakan wireshark. Hasil pengujian parameter *throughput* untuk tiap topologi dapat dilihat pada Tabel 4.1. Pembahasan akan dilakukan pada bab berikutnya.

Tabel 5.1.1 Pengujian Parameter Throughput

Topologi	Jumlah Node	Jumlah Paket Terkirim	Time span (s)	Average pps	Average packet size, B	Bytes	Average Bytes/s	Average Bits/s (Throughput)
Grid	10	155 (3,6%)	238,916	0,6	84	13020 (3,9%)	54	435
	20	212 (3,4%)	572,865	0,4	84	17808 (3,7%)	31	248
	40	305 (3,0%)	1679,690	0,4	84	28391 (3,3%)	21	152
	80	357 (2,5%)	3098,256	0,4	84	33231 (2,9%)	17	102
Random	10	150 (3,3%)	239,381	0,6	84	12600 (3,6%)	52	421
	20	207 (3,1%)	1271,404	0,2	84	17388 (3,4%)	28	219

	40	301 (2,7%)	3089 ,912	0,2	84	37327 (3,0%)	18	113
	80	355 (2,2%)	3217 ,821	0,2	84	40306 (2,4%)	12	89

5.1.2 Hasil Delay

Pada sub bab ini ditampilkan hasil pengujian terhadap parameter *delay*. Pengujian ini melibatkan 10, 20, 40, dan 80 node pada topologi grid dan topologi random dengan dilakukan selama 500 detik. Selama pengujian berlangsung radio messages tools akan melakukan packet capture yang menghasilkan file .pcap untuk kemudian dianalisis menggunakan wireshark. Hasil pengujian parameter *delay* untuk tiap topologi dapat dilihat pada Tabel 4.2. Pembahasan akan dilakukan pada bab berikutnya.

Tabel 5.2 Pengujian Parameter Delay

Topologi	Jumlah Node	Jumlah Paket Terkirim	Total delay (ms)	Average Delay (ms)
Grid	10	155 (3,6%)	26,986	174
	20	212 (3,4%)	62,914	297
	40	305 (3,0%)	96,756	317
	80	357 (2,5%)	143,327	401
Random	10	150 (3,3%)	45,998	307
	20	207 (3,1%)	78,692	380
	40	301 (2,7%)	102,559	340
	80	355 (2,2%)	170,041	479

5.1.3 Hasil Jitter

Pada sub bab ini ditampilkan hasil pengujian terhadap parameter *jitter*. Pengujian ini melibatkan 10, 20, 40, dan 80 node pada topologi grid dan topologi random dengan dilakukan selama 500 detik. Selama pengujian berlangsung radio messages tools akan melakukan packet capture yang menghasilkan file .pcap untuk kemudian dianalisis menggunakan wireshark. Hasil pengujian parameter *jitter* untuk tiap topologi dapat dilihat pada Tabel 4.3. Pembahasan akan dilakukan pada bab berikutnya.

Tabel 5.3 Pengujian Parameter Jitter

Topologi	Jumlah Node	Jumlah Paket Terkirim	Total delay variation (ms)	Jitter (ms)
Grid	10	155 (3,5%)	278	1,795

Random	20	212 (3,4%)	649	3,06
	40	305 (3,0%)	997	3,271
	80	357 (2,5%)	1478	4,14
	10	150 (3,3%)	474	3,162
Random	20	207 (3,1%)	811	3,92
	40	301 (2,7%)	1057	3,514
	80	355 (2,2%)	1753	4,94
	10	150 (3,3%)	474	3,162

5.2 Analisis

Sub bab ini akan membahas penjelasan dan analisis dari pengujian yang telah dilakukan sebelumnya. Analisis meliputi perbandingan tiap parameter QoS yang telah ditentukan sebelumnya terhadap tiap topologi dengan jumlah node berbeda-beda. Parameter QoS yang diuji adalah *throughput*, *delay*, dan *jitter*. Analisis disajikan dalam bentuk Tabel dengan memberikan perbandingan antar tiap topologi dengan jumlah node yang berbeda-beda.

5.2.1 Analisis Perbandingan *Throughput*

Analisis Perbandingan *throughput* berikut dilakukan setelah pengujian selama 500 detik dijalankan pada simulasi sesuai dengan skenario awal. Setelah file.pcap dari radio message tools, file tersebut lalu diolah menggunakan wireshark.

Data hasil pengujian secara umum tiap topologi dengan node yang berbeda-beda untuk parameter QoS *throughput* dapat dilihat pada Tabel 5.1 berikut.

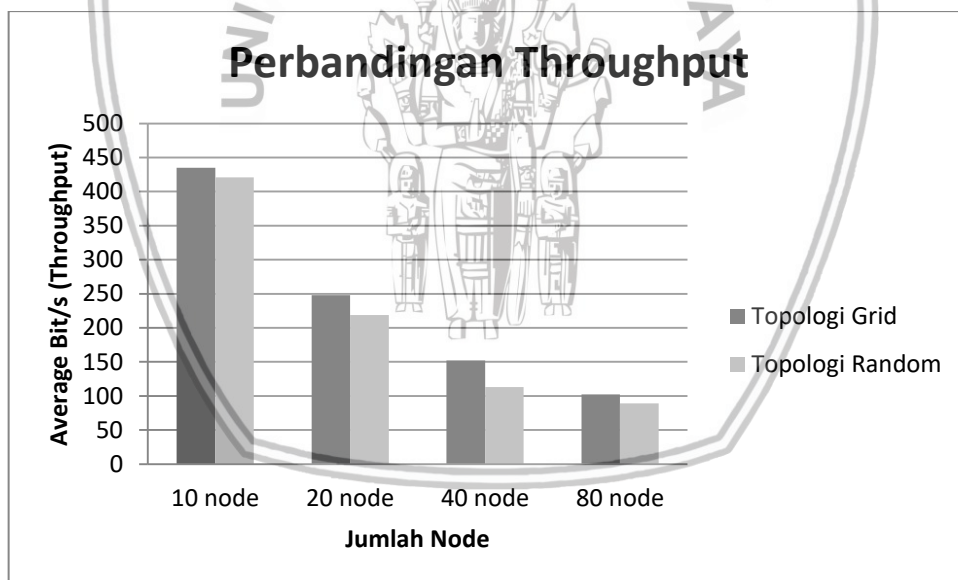
Tabel 5.4 Analisis Perbandingan *Throughput*

Topologi	Jumlah Node	Jumlah Paket Terkirim	Time span (s)	Average pps	Average packet size, B	Bytes	Average Bytes/s	Average Bits/s (Throughput)
Grid	10	155 (3,6%)	238,916	0,6	84	13020 (3,9%)	54	435
	20	212 (3,4%)	572,865	0,4	84	17808 (3,7%)	31	248
	40	305 (3,0%)	1679,690	0,4	84	28391 (3,3%)	21	152
	80	357 (2,5%)	3098,256	0,4	84	33231 (2,9%)	17	102
Random	10	150 (3,3%)	239,381	0,6	84	12600 (3,6%)	52	421

	20	207 (3,1%)	1271 ,404	0,4	84	17388 (3,4%)	28	219
	40	401 (2,7%)	3089 ,912	0,2	84	37327 (3,0%)	18	113
	80	433 (2,2%)	3217 ,821	0,2	84	40306 (2,4%)	12	89

Data pada Tabel 5.1 ini diperoleh dengan memfilter file.pcap menggunakan ip address tujuan dan protokol IPv6 sehingga diperoleh jumlah paket, lama waktu, rata-rata paket per detik, rata-rata ukuran paket dalam bytes dan rata-rata Throughput.

Berdasarkan Tabel 5.1 rata-rata *throughput* paling tinggi dengan 435 bit/s adalah Topologi grid dengan 10 node, disusul oleh topologi random 10 node dengan 421 bit/s, lalu secara berurutan topologi grid 20 node dengan 248 bit/s, topologi random 20 node dengan 219 bit/s, topologi grid 40 node dengan 152 bit/s, topologi random 40 node dengan 113 bit/s, topologi grid 80 node dengan 102 bit/s, dan yang terakhir topologi random 80 node dengan 89 bit/s.



Gambar 5.1 Analisis Perbandingan Throughput

Dapat dilihat pada Gambar 5.1 rata-rata *throughput* pada topologi grid lebih tinggi daripada topologi random pada setiap node yang berbeda-beda. Rata-rata Throughput paling tinggi dengan 435 bit/s adalah Topologi grid dengan 10 node, sedangkan rata-rata *throughput* paling rendah dengan 89 bit/s adalah topologi random dengan 80 node.

Pada Tabel 5.4 dapat dilihat rata rata paket per detik topologi grid dan topologi random pada jumlah node 10 dan 20 tidak berbeda yaitu 0,6 dan 0,4 paket. Pada kolom time span dalam detik antara topologi grid dan

topologi random pada jumlah node 10 dan 20 memiliki perbedaan, dimana time span pada topologi random dengan jumlah node 10 dan 20 adalah 239,381 detik dan 1271,404 detik sementara pada topologi grid dengan jumlah node 10 dan 20 adalah 238,916 detik dan 572,865 detik. Dengan asumsi bahwa semua paket diterima oleh tujuan paket, maka hal ini akan menentukan perolehan *throughput* dimana lama waktu paket yang diterima merupakan parameter tertentu.

Pada Gambar 5.1 *throughput* mengalami penurunan saat jumlah node yang bertambah. Hal tersebut terjadi karena semakin bertambahnya jumlah node maka mobilitas dalam pengiriman paket tidak lagi stabil. Karena routing RPL memilih jarak terdekat node receiver dengan node sender dalam pengiriman paket, maka dari itu semakin banyak jumlah node sender maka node receiver akan mencari jarak terdekat dari jumlah node sender tersebut. Apalagi jika salah satu node sender mengirim paket ke node receiver memiliki waktu yang lama maka antrian node sender selanjutnya juga akan semakin lama sehingga *average throughput* akan memburuk.

5.2.2 Analisis Perbandingan Delay

Analisis perbandingan *delay* berikut dilakukan setelah pengujian selama 500 detik dijalankan pada simulasi sesuai dengan skenario awal. Setelah file info waktu diperoleh dari motes output, file tersebut di filter secara manual, lalu data tersebut di masukkan ke MS Excel sehingga waktu *delay* dapat diperoleh.

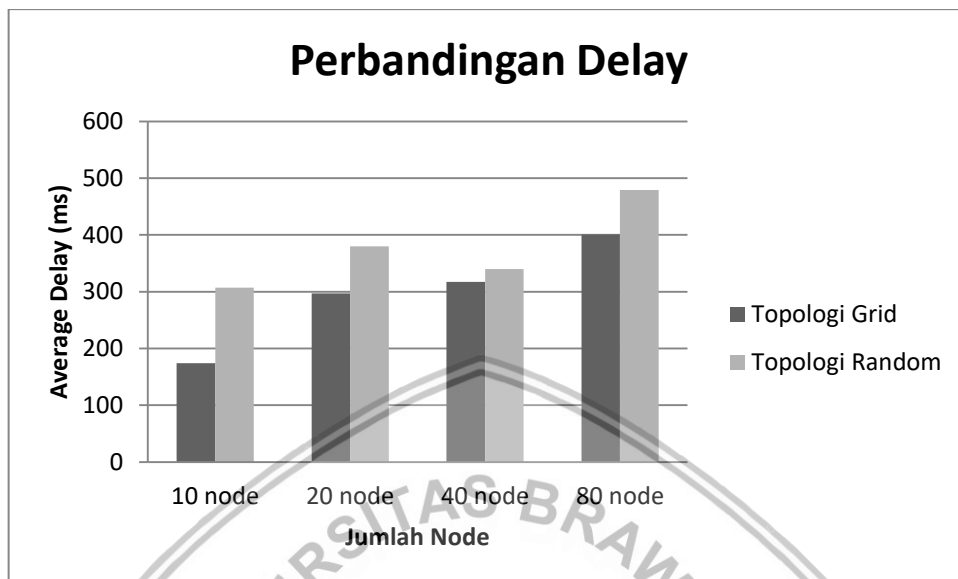
Data hasil pengujian secara umum tiap topologi dengan node yang berbeda-beda untuk parameter QoS *delay* dapat dilihat pada Tabel 5.2 berikut.

Tabel 5.5 Analisis Perbandingan Delay

Topologi	Jumlah Node	Jumlah Paket Terkirim	Total delay (ms)	Average Delay (ms)
Grid	10	155 (3,6%)	26,986	174
	20	212 (3,4%)	62,914	297
	40	305 (3,0%)	96,756	317
	80	357 (2,5%)	143,327	401
Random	10	150 (3,3%)	45,998	307
	20	207 (3,1%)	78,692	380
	40	301 (2,7%)	102,559	340
	80	355 (2,2%)	170,041	479

Berdasarkan Tabel 5.2 rata-rata *delay* terendah diperoleh pada topologi grid 10 node dengan 174 ms, disusul secara berurutan topologi grid 20 node dengan 297 ms, topologi random 10 node dengan 307 ms, topologi

grid 40 node dengan 317 ms, topologi random 40 node dengan 340 ms, topologi random 20 node dengan 380 ms, topologi grid 80 node dengan 401 ms, dan yang terakhir topologi random 80 node dengan 479 ms.



Gambar 5.2 Analisis Perbandingan Delay

Dapat dilihat pada Gambar 5.2 secara rata-rata topologi grid lebih baik daripada topologi random dalam perolehan *delay* tiap jumlah node yang berbeda-beda. Hal ini disebabkan jarak antar node pada topologi grid lebih berdekatan daripada topologi random pada area 100x100 meter, sedangkan rata-rata *delay* pada topologi grid dengan jumlah node 20 adalah 297 ms sementara itu pada topologi random dengan jumlah node 10 adalah 307 ms. Sama halnya rata-rata *delay* pada topologi grid dengan jumlah node 40 adalah 317 ms sementara pada topologi random dengan jumlah node 20 adalah 380 ms. Hal ini disebabkan pada topologi grid jarak node sender saling berdekatan pada jarak yang sama dengan node. Sedangkan pada topologi random jarak node sender dengan node receiver ada yang saling berdekatan dan ada yang saling berjauhan. Dari hal ini perolehan rata-rata *delay* dapat diperoleh dengan baik apabila letak node sender berdekatan dengan node receiver.

5.2.3 Analisis Perbandingan Jitter

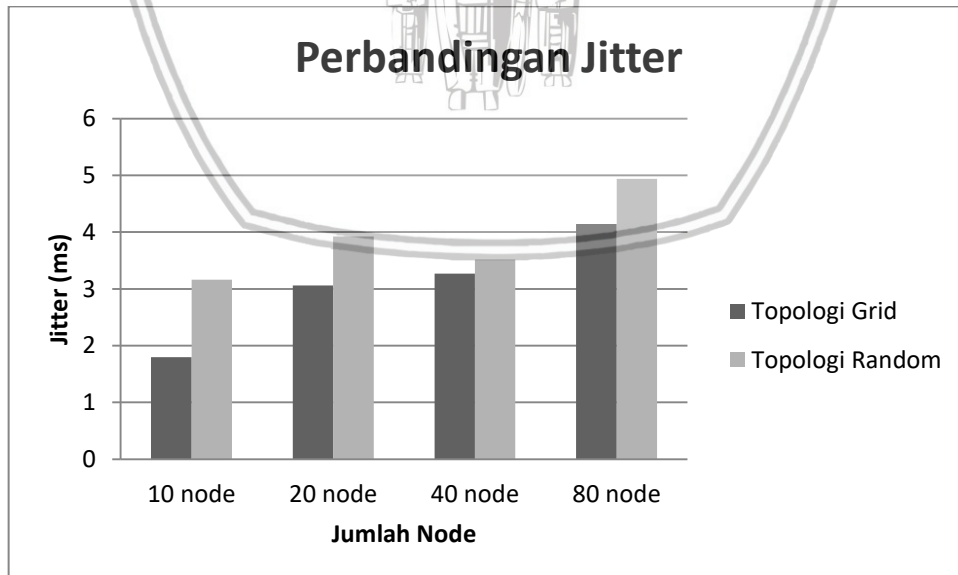
Analisis Perbandingan *jitter* berikut dilakukan setelah pengujian selama 500 detik dijalankan pada simulasi sesuai dengan skenario awal. Setelah file info waktu diperoleh dari motes output, file tersebut di filter secara manual, lalu data tersebut di masukkan ke MS Excel sehingga waktu *delay* dapat diperoleh. Jitter merupakan variasi dari delay maka dibutuhkan perhitungan delay sebelumnya yang digunakan kembali untuk mencari nilai *delay variation* yang dapat digunakan untuk menghitung *jitter*.

Data hasil pengujian secara umum tiap topologi dengan node yang berbeda-beda untuk parameter QoS Jitter dapat dilihat pada Tabel 5.3 berikut.

Tabel 5.6 Analisis Perbandingan *Jitter*

Topologi	Jumlah Node	Jumlah Paket Terkirim	Total delay variation (ms)	Jitter (ms)
Grid	10	155 (3,5%)	278	1,795
	20	212 (3,4%)	649	3,06
	40	305 (3,0%)	997	3,271
	80	357 (2,5%)	1478	4,14
Random	10	150 (3,3%)	474	3,162
	20	207 (3,1%)	811	3,92
	40	301 (2,7%)	1057	3,514
	80	355 (2,2%)	1753	4,94

Berdasarkan Tabel 5.3 *jitter* terendah adalah topologi grid 10 node dengan hanya 1,795 ms, lalu secara berurutan topologi grid 20 node dengan 3,06 ms, topologi random 10 node dengan 3,162 ms, topologi grid 40 node dengan 3,271 ms, topologi random 40 node dengan 3,514 ms, topologi random 20 node dengan 3,92 ms, topologi grid 80 node dengan 4,14 ms, dan yang terakhir topologi random 80 node dengan 4,94 ms.



Gambar 5.3 Analisis Perbandingan *Jitter*

Dapat dilihat pada Gambar 5.3 perolehan *jitter* pada topologi random lebih baik daripada topologi grid. *Jitter* pada topologi grid dengan jumlah node 20 adalah 3,06 ms sementara itu pada topologi random dengan jumlah

node 10 adalah 3,162 ms. Sama halnya *jitter* pada topologi grid dengan jumlah node 40 adalah 3,271 ms sementara pada topologi random dengan jumlah node 20 adalah 3,92 ms.

Dapat dilihat bahwa pada topologi grid dengan jumlah node 20 dan 40 sedangkan pada topologi random dengan jumlah node 10, 20 dan 40 memperoleh *jitter* yang sama dengan kisaran 3 ms. Grafik pada perbandingan *jitter* pada Gambar 5.3 dibidang sama dengan grafik pada perbandingan *delay* pada Gambar 5.2. Hal ini disebabkan sama dengan perolehan rata-rata *delay* yaitu pada topologi grid jarak node sender saling berdekatan pada jarak yang sama dengan node. Sedangkan pada topologi random jarak node sender dengan node receiver ada yang saling berdekatan dan ada yang saling berjauhan. Dari hal ini perolehan rata-rata *delay* dapat diperoleh dengan baik apabila letak node sender berdekatan dengan node receiver. Sedangkan total variasi *delay* diperoleh dengan menghitung selisih *delay* pada topologi dengan jumlah node tertentu, sementara rata-rata *delay* diperoleh dari total delay yang di rata-rata kan.



BAB 6 KESIMPULAN

Pada bab ini berisi kesimpulan serta saran terhadap hasil kinerja protokol 6LoWPAN pada jaringan sensor nirkabel yang disimulasikan pada *Cooja Simulator*.

6.1 Kesimpulan

Dari hasil pengujian dan analisis kinerja protokol 6LoWPAN dengan topologi grid dan topologi random pada parameter *throughput*, *delay*, dan *jitter*, dapat diimpulkan bahwa :

1. Lingkungan simulasi berbasis 6LoWPAN dapat dibangun pada *Cooja simulator*, dengan menentukan terlebih dahulu jenis node yang dibutuhkan, dalam hal ini *sender node* dan *receive node*. Kemudian node yang disimulasikan dimasukkan kedalam simulasi. Pada penelitian ini jenis mode yang disimulasikan adalah *skymotes*. Jenis mode ini dipilih karena mendukung protokol 6LoWPAN untuk melakukan simulasi.
2. Arsitektur jaringan sensor nirkabel diterapkan dengan menggunakan topologi yang bisa disimulasikan menggunakan *cooja simulator*. Topologi yang dibuat ada 2 yaitu topologi grid dan topologi random yang masing mempunyai 10, 20, 40, dan 80 *sender node*. Setiap topologi terdapat satu *receive node*.
3. Analisis *throughput*, *delay* dan *jitter* dilakukan menggunakan *Wireshark* dan keluaran *mote output* setelah simulasi jaringan WSN dijalankan pada *cooja simulator*. Hasil simulasi berupa *file.pcap* dan *file.txt*. Parameter *throughput* menggunakan *file.pcap* yang dibuka dengan *Wireshark*, lalu dianalisis menggunakan fitur *capture file properties*, sementara untuk menganalisis *delay* dan *jitter* dilakukan secara manual menggunakan *file.txt*, lalu di hitung secara manual menggunakan *MS office Excel*.
4. Berdasarkan analisis yang dilakukan secara rata-rata *Throughput* pada topologi grid lebih tinggi daripada topologi random pada setiap node yang berbeda-beda. Rata-rata *Throughput* paling tinggi dengan 435 bit/s adalah Topologi grid dengan 10 node, sedangkan rata-rata *Throughput* paling rendah dengan 89 bit/s adalah topologi random dengan 80 node. Sedangkan rata-rata *delay* topologi grid lebih baik daripada topologi random dalam perolehan *delay* tiap jumlah node yang berbeda-beda. Hal ini disebabkan jarak antar node pada topologi random lebih berdekatan daripada topologi grid pada area 100x100 meter, pada topologi grid lebih baik daripada topologi random disebabkan jarak node sender saling berdekatan dengan node receiver. Dan pada rata-rata *jitter* topologi random lebih baik daripada topologi grid. Padahal topologi grid dan topologi random dengan jumlah node yang berbeda-beda dalam perolehan *jitter* dapat terbilang hampir sama atau tidak jauh berbeda dengan yang lain tetapi terdapat nilai *jitter* yang berbeda. Pada grafik perbandingan *jitter* sama dengan grafik perbandingan *delay*.

6.2 Saran

Saran yang bisa penulis sampaikan untuk penelitian lebih lanjut terkait dengan penelitian ini adalah:

1. Diperlukan pengujian lebih lanjut tentang analisis kinerja protokol 6LoWPAN pada jaringan nirkabel
2. Diperlukan pengujian dengan menggunakan parameter-parameter lainnya.
3. Skenario pengujian dapat di kembangkan dengan yang lebih bervariasi.



DAFTAR PUSTAKA

- Amer, N., 2012. Comparison study between IPV4 & IPV6 Philadelphia University, Jordan, CIS departmen
- Anggraini, D., Irawati, I. D., dan Mayasari, R., 2014, Analisis Dan Simulasi Wireless Sensor Network (WSN) Untuk Komunikasi Data Menggunakan Protokol Zigbee, Bandung: Telkom University
- Arrosyid, M.H., Tjahjono, A., dan Sunarno, E., 2009. Implementasi Wireless Sensor Network Untuk Monitoring Parameter Energi Listrik Sebagai Peningkatan Layanan Bagi Penyedia Energi Listrik. Institut Teknologi Sepuluh November
- Carvalho, F., dan Magedanz, T., 2009. Telecommunication Systems and Technologies – VOL II - Quality of Service in Telecommunication Networks. Encyclopedia of Life Support Systems (EOLSS)
- Garg, N., 2015. Network Simulators: A Case Study. University College Kurukshetra University, Kurukshetra, Haryana, India
- Jiang, Y., Zhang, L., Wang, L. (2013). Wireless sensor networks and the internet of things. International Journal of Distributed Sensor Networks, 2013
- Lewis, F.L. 2004. Wireless Sensor Network. University of Texas.
- Nurul, H.A.I., Hassan, R., Khadijah, W.M., dan Ghazali. 2012 A Study On Protocol Stack In 6lowpan Model. Universiti Kebangsaan Malaysia
- Ross, K., dan Kurose, J. 2013. *Computer Networking ; a top-down approach*. 6 penyunt. New Jersey: Pearson Education, Inc.
- Roussel, K., Ye-Qiong S, dan Zendra O., 2016. Using Cooja for WSN Simulations: Some New Uses and Limits. HAL
- Satrio, A., Munadi, R., dan Mayasari, R., 2016. Analisis Pengaruh Pola Penyebaran Node Pada Jaringan Sensor Nirkabel, Bandung: Telkom University
- Schonwalder, J., 2010. Internet of things: 802.15.4, 6LoWPAN, RPL, COAP.
- Shelby, Z., dan Bormann, C. 2009, 6LoWPAN: The Wireless Embedded Internet. John Wiley & Sons Ltd. United Kingdom
- Tama, dan Putra, H., 2010. Perancangan dan Impelementasi Wireless Sensor Network (WSN) Berbasis IEEE 802.15.4/Zigbee. Universitas Komputer Indonesia, Bandung, Indonesia.
- Toscano, dan Lo Bello. 2012. Comparative assessments of IEEE 802.15.4/Zigbee and 6LoWPAN for low-power industrial WSNs in realistic scenarios. Factory Communication System (WFCS)
- TR 101 329 V2.1.1. 1999. *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON)*; General aspects of Quality of Service (QoS)

- Tsvetkov, T., 2011. RPL: IPv6 Routing Protocol for Low Power and Lossy Networks. Technische Universitat Munchen
- Vasseur, J.P., Agarwal, N., Hui, J., Shelby, Z., Bertrand, P., dan Chauvenet, C., 2011. RPL: The IP routing protocol designed for low power and lossy networks. Internet Protocol for Smart Objects (IPSO) Alliance
- Vucinic, M., dan Romaniello, G., 2014. Topology Construction in RPL Networks over Beacon-Enabled 802.15.4. Grenoble Alps University

